

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»
Профиль подготовки: «Программное обеспечение средств вычислительной
техники и автоматизированных систем»

БАКАЛАВРСКАЯ РАБОТА

на тему

**«ПРОГРАММА СОСТАВЛЕНИЯ
ОТЧЁТНОСТИ МАЙНИНГ-СТРУКТУРЫ»**

Студент Матросов 20.05.26 Матросов Александр Михайлович
(подпись, дата) (ФИО полностью)

Руководитель Леонова 20.05.26 Леонова Т.Ю.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр Бычков 25.05.26 Бычков А.С.
(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 22.05.2026 г. № 14)

Заведующий кафедрой Митрохин М.А. Митрохин
(подпись) (фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от
14 июня 2026 г. № 24)

Секретарь ГЭК Гурин Е.И. Гурин
(подпись) (фамилия, инициалы)

Пенза 2026

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

 «Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«07» апрель 2026 г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Матросов Александр Михайлович гр. 22ВВП1 факультета ФВТ

2. Тема работы Программа составления отчётности майнинг-структуры

Тема утверждена приказом ПГУ № 347/о от " 07 " апреля 2026 г.

3. Руководитель бакалаврской работы Леонова Т.Ю.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

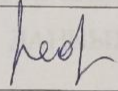
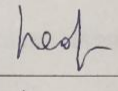
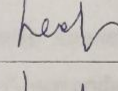
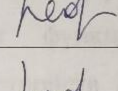
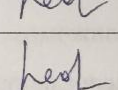
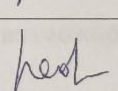
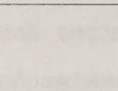
Разработать программу для ускорения выполнения подготовки отчетности майнинг-центра. Тема является актуальной в условиях роста спроса на криптовалютный рынок и на всю экосистему по добыче виртуальных монет. Для того чтобы достичь максимальной энергоэффективности при работе устройств, а также снизить риск серьезных изменений в майнинг-центре, часто применяют системы мониторинга, которые позволяют собирать отчетность по всем устройствам.

Разработанная программа для ускорения выполнения подготовки отчетности в майнинг-центре позволяет уменьшить расходы в производстве, избавиться от лишних ошибок и сконцентрироваться на работе устройств, отслеживать всю необходимую информацию для повышения энергоэффективности самих устройств и сокращения времени на выполнение рутинной задачи.

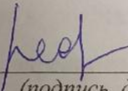
5. Перечень подлежащих разработке вопросов

Требуется произвести анализ предметных областей: изучить существующие информационные системы, выделить их достоинства и недостатки. Выявить потребности пользователей. Выбрать язык программирования, среду разработки, изучить вопрос электротехники и схемотехники. Определить архитектуру приложения и вспомогательные средства разработки. Изучить функционал Visual Studio для разработки программы для ускорения выполнения подготовки отчетности майнинг-центра.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Изучение предметной области	32	07.04.2026 – 10.04.2026	
2	Анализ требований	32	11.04.2026 – 14.04.2026	
3	Создание макета графического интерфейса	30	15.04.2026 – 19.04.2026	
4	Выбор технологий, применяемых при проектировании и разработке	40	20.04.2026 – 25.04.2026	
5	Реализация графического интерфейса	66	26.04.2026 – 04.05.2026	
6	Разработка основной части программы	70	05.05.2026 – 12.05.2026	
7	Оформление пояснительной записки	54	13.05.2026 – 21.05.2026	

Дата выдачи задания "07" апреля 2026 г.

Руководитель бакалаврской работы  07.04.2026
(подпись, дата) Леонова Т.Ю.

Задание к исполнению принял студент  07.04.2026
(подпись, дата) Матросов А.М.

Работу к защите допустить
Декан факультета  Фионова Л.Р.
(подпись, дата) 03.06.2020

Реферат

Бакалаврская работа содержит 115 листов, 39 рисунков, 1 таблицу, 7 источников, 3 приложения.

ОПЕРАТИВНОСТЬ, УСКОРЕНИЕ, АНАЛИЗ, МАЙНИНГ-ЦЕНТР, ПАРСИНГ, WEB, ОТЧЕТНОСТЬ, ПОЛЬЗОВАТЕЛЬ, СБОР ДАННЫХ, ПРОГРАММИРОВАНИЕ, ЭФФЕКТИВНОСТЬ

Цель работы – разработать программу для ускорения выполнения подготовки отчетности майнинг-центра.

Одним из ключевых факторов выполнения основной задачи является разработка программы сбора и мониторинга данных через веб-интерфейс платформы Pitbit, обеспечивающая выполнение следующих функций: автоматизация процесса получения технических показателей с устройств для майнинга, простота в эксплуатации, стабильность работы системы, возможность анализа и экспорта данных в структурированном виде, использование современных инструментов разработки на платформе .NET, возможность расширения функционала программы новыми модулями.

В конечном итоге разработано приложение для операционной системы Windows, которое позволяет настраивать эффективный сбор, отображение и анализ данных с устройств для майнинга, зарегистрированных на платформе Pitbit. Данная программа полностью соответствует цели данной выпускной квалификационной работы. В ходе достижения результата были использованы навыки объектно-ориентированного программирования на языке C#, работы с библиотеками автоматизации браузера, парсинга HTML и создания отчетов. Данная программа протестирована и готова к эксплуатации.

					ПГУ1.090301.18.001 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	«Программа составления отчётности майнинг-структуры» Пояснительная записка	Лит.	Лист	Листов	
Разраб.		Матросов А.М.		20.05.20					
Провер.		Леонова Т.Ю.		22.05.20			2	115	
Н. Контр.		Бычков А.С.		25.05.20		ФВТ, каф. ВТ гр. 22ВВП1			
Утверд.		Митрохин М.А.		25.05.20					

Содержание

Перечень сокращений и обозначений	5
Введение.....	7
1 Программа составления отчётности майнинг-структуры.....	9
1.1 Анализ предметной области	9
1.2 Обзор аналогов	10
1.3 Анализ требований к разработке	13
1.4 Выбор архитектуры системы	14
1.5 Выбор языка программирования	20
1.6 Выбор среды разработки.....	21
1.7 Проектирование и разработка	29
1.8 Разработка управляющей программы.....	31
1.8.1 Инициализация пользовательского интерфейса	32
1.8.2 Авторизация на платформе Pitbit	33
1.8.3 Сбор данных об устройствах	34
1.8.4 Отображение и фильтрация данных	35
1.8.5 Экспорт данных в Excel	36
1.8.6 Экспорт в формат hash-реестра	37
1.8.7 Загрузка данных из Excel.....	38
1.8.8 Вспомогательные функции и обработка событий	39
1.8.9 Архитектура развертывания.....	39
1.9 Графический интерфейс.....	40
1.9.1 Сценарий «Авторизация в системе»	41
1.9.2 Сценарий «Сбор данных об устройствах»	42
1.9.3 Сценарий «Переключение режима отображения»	43

1.9.4	Сценарий «Просмотр детальной информации об устройстве»	44
1.9.5	Сценарий «Экспорт данных в Excel»	45
1.9.6	Сценарий «Экспорт в Hash-реестр»	47
1.9.7	Сценарий «Загрузка данных из Excel»	48
1.9.8	Сценарий «Очистка данных»	50
1.9.9	Дополнительные возможности	50
2	Требования безопасности по охране труда при работе с ПЭВМ.....	52
2.1	Организация рабочего места	53
2.2	Освещение рабочего места	54
2.3	Электробезопасность	55
2.4	Пожарная безопасность	56
2.5	Режим труда и отдыха	57
	Заключение	60
	Список использованных источников	61
	Приложение А – UML диаграммы и схемы	63
	Приложение Б – Листинг программ	70
	Приложение В – Презентация	112

Перечень сокращений и обозначений

ПЭВМ (персональная электронно-вычислительная машина) — термин, используемый в разделе требований безопасности для обозначения персонального компьютера

ПО (программное обеспечение) — обобщающий термин для компьютерных программ. Используется в разделе обзора аналогов и в разделе требований безопасности

API (Application Programming Interface) — набор правил и инструментов для взаимодействия между различными программными компонентами. В работе используется для описания возможности интеграции программы с внешними сервисами

ASIC (Application-Specific Integrated Circuit) — тип оборудования для майнинга криптовалют, предназначенный для выполнения конкретных вычислительных задач. В тексте упоминается применительно к майнинговым устройствам и платформам мониторинга

GPU (Graphics Processing Unit) — тип вычислительного устройства, используемый для майнинга криптовалют наряду с ASIC. Упоминается в сравнительной таблице аналогов

HTTPS (HyperText Transfer Protocol Secure) — протокол для безопасного обмена данными между клиентом и сервером. В работе указан как протокол взаимодействия программы с удалённым сервером Pitbit.Online

IDE (Integrated Development Environment) — программное обеспечение, предоставляющее инструменты для написания, отладки и компиляции кода. В тексте упоминается применительно к Microsoft Visual Studio 2022

IP-адрес (Internet Protocol Address) — уникальный идентификатор устройства в компьютерной сети, работающей по протоколу IP. Используется в таблице данных о майнерах для идентификации оборудования

NuGet — менеджер пакетов для платформы .NET. Система, позволяющая подключать сторонние библиотеки к проекту. Упоминается в разделе о выборе среды разработки

UML (Unified Modeling Language) — язык графического описания для проектирования программных систем. В тексте упоминаются UML-диаграммы использования и последовательностей

					<i>ПГУ1.090301.18.001 ПЗ</i>	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

Введение

Необходимость ускорения процесса сбора и оформления отчетности имеет место быть практически в любой сфере деятельности человека. В зависимости от некоторых условий требуется большое количество времени на выполнение рутинной задачи, которую можно выполнить путем реализации решений с технологическим подходом. Тема криптовалют и майнинга становится все более популярными в наше время. И она набирает обороты не на уровне обывателя, а в больших промышленных масштабах. В связи с этим появляются технические проблемы, будь то ремонт оборудования или же сбои в работе системы. Ведь одной из важнейших характеристик майнинг — системы является ее скорость. И именно отслеживание работы устройств позволяет определить, какие аспекты можно изменить в ходе работы самих машин для майнинга.

Чтобы избежать данных проблем, одним немаловажным решением будет создать программу, которая позволит создавать отчетность по работе самой системы, снизить расходы на персонал и проследить за качеством работы системы от различных ситуаций. Человек во время управления техническим устройством должен рассматриваться как единая система. Основной задачей управления данной системой заключается в преобразовании команд оператора в соответствующие воздействия, а также просмотр информации о состоянии системы.

Эффективность обратной связи напрямую зависит от оперативности и достоверности собираемых данных. В условиях промышленного майнинга, где информация о хешрейте, температуре и энергопотреблении каждого отдельного устройства непрерывно генерируется и отображается на специализированных веб-платформах, возникает задача ее агрегации для последующего анализа. Существующая практика ручного копирования этих данных в табличные процессоры, такие как Excel, является узким местом, снижающим общую эффективность системы управления.

					ПГУ1.090301.18.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

Целью данной работы является разработка инструмента, автоматизирующего процесс сбора и первичной обработки телеметрии оборудования. Создаваемая программа реализует парсинг данных с веб-интерфейса системы мониторинга майнинг-фермы и их автоматическую вставку в структурированный шаблон Excel. Такой подход позволяет унифицировать процесс формирования отчетности, исключить влияние человеческого фактора на этапе сбора данных и предоставить оператору и руководству консолидированную информацию в привычном и удобном для дальнейшей обработки формате.

					ПГУ1.090301.18.001 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

1 Программа составления отчётности майнинг-структуры

1.1 Анализ предметной области

Процесс промышленного майнинга криптовалют представляет собой сложную производственную деятельность, в рамках которой задействовано дорогостоящее специализированное оборудование, функционирующее в круглосуточном режиме с высокими нагрузками на вычислительные модули, системы охлаждения и электропитания, при этом в ходе непрерывной эксплуатации неизбежно возникают внештатные ситуации: деградация отдельных чипов, падение хэшрейта, перегрев плат, скачки энергопотребления и потеря связи с пулом, что требует оперативного вмешательства технического персонала для предотвращения полного выхода устройства из строя и, как следствие, прямых финансовых потерь от простоя мощностей. В условиях масштабирования майнинговых ферм и перехода к индустриальной модели ведения бизнеса количество одновременно обслуживаемых единиц техники исчисляется десятками и сотнями, а стандартные средства веб-мониторинга пулов, ориентированные на ручной просмотр и визуальный анализ табличных данных, создают критическую информационную перегрузку оператора, повышая вероятность пропуска ранних признаков неисправности из-за так называемого «человеческого фактора» — усталости, отвлечения внимания или банальной ошибки при переносе показателей в отчётные формы.

Разработанная управляющая программа решает задачу автоматизации сбора, первичной обработки и структурирования данных о состоянии оборудования, обеспечивая устойчивость к информационным перегрузкам и исключая влияние субъективных ошибок персонала за счёт прямого асинхронного взаимодействия с веб-интерфейсом мониторинга и файловой системой отчётности. Накопление систематизированной аналитики в едином хранилище позволяет не только оперативно реагировать на текущие

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

отклонения в работе устройств, но и формировать историческую базу показателей производительности и отказов, на основе которой возможно прогнозирование сроков выхода оборудования из строя, планирование закупок запасных частей и комплектующих, оптимизация графиков технического обслуживания и перераспределение вычислительных нагрузок между фермами.

С точки зрения бизнеса, внедрение подобной системы обеспечивает переход от реактивной модели управления, при которой действия предпринимаются уже по факту случившейся поломки, к проактивной стратегии, направленной на превентивное выявление рисков и минимизацию времени простоя оборудования. Кроме того, высвобождение временных и человеческих ресурсов, ранее затрачиваемых на монотонную рутинную работу по ручному заполнению таблиц и отслеживанию изменений характеристик, позволяет перераспределить усилия персонала на решение более приоритетных производственных задач — непосредственный ремонт вышедших из строя узлов, модернизацию систем охлаждения и электропитания, а также расширение парка оборудования, что в конечном счёте способствует повышению общей рентабельности, отказоустойчивости и конкурентоспособности майнинг-центра в условиях нарастающей индустриализации отрасли.

1.2 Обзор аналогов

Готовых решений не очень много, так как данная сфера только набирает популярность. Отечественные разработки отсутствуют на рынке.

Есть зарубежные исполнения, но только для устройств с воздушным охлаждением и меньшим функционалом.

«Minerstat» — данный сервис представлен на рисунке 1. Minerstat является частью многоуровневого сервиса, предназначенного для удаленного управления и мониторинга майнинг установок криптоэнтузиастов. Minerstat

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

не является сервисом облачного майнинга или посредником, он только предоставляет ПО. Клиенты работают на выбранных ими пулах, и от них получают монеты, которые там добыли. Сервис не ориентирован на новичков. Нужны базовые знания о том, как использовать пулы, генерировать кошельки, и настраивать конфигурации.



Рисунок 1 — «Minerstat»

«Vixbit» — это продвинутая платформа для майнинга криптовалют, предоставляющая облачный мониторинг, индивидуальные прошивки и решения для управления аппаратным обеспечением для ASIC-майнинговых устройств. Vixbit специализируется на решениях для управления операциями по майнингу криптовалют с высокой производительностью. Основное предложение включает автоматизированную систему мониторинга — облачную платформу, позволяющую майнерам мониторить, управлять и оптимизировать ASIC устройства в реальном времени. Это обеспечивает эффективную работу майнинга, минимизацию простоев и своевременное уведомление о заранее определённых событиях. Страницу сайт можно посмотреть на рисунке 2 [1].

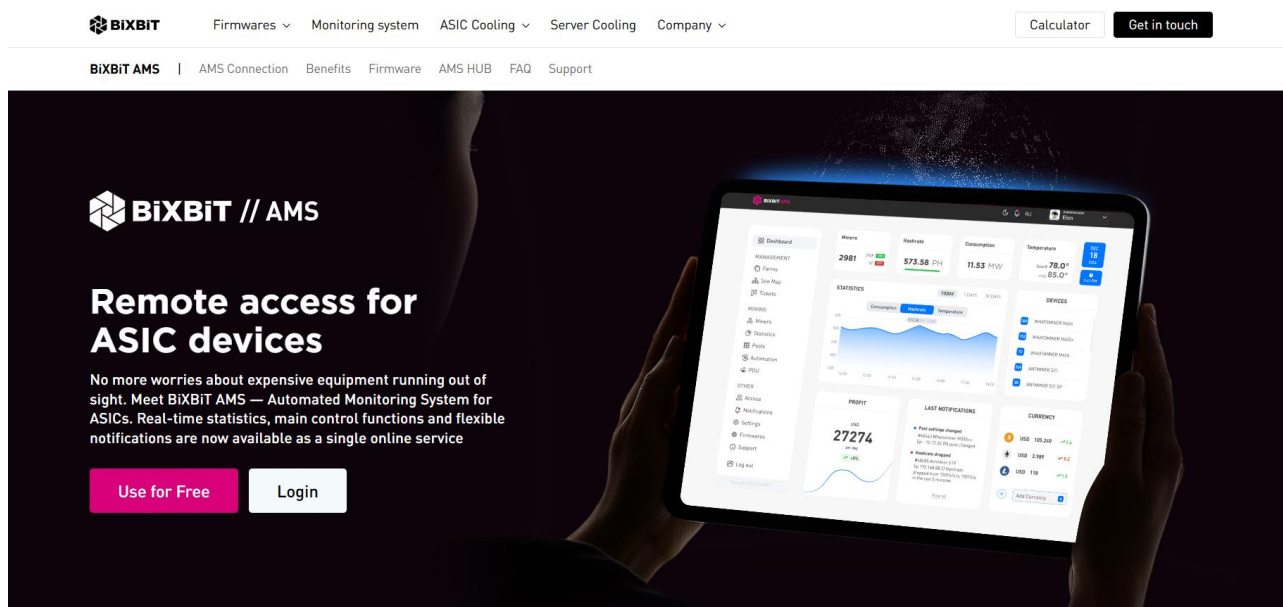


Рисунок 2 — Vixbit

Monito.run — это платформа для мониторинга, управления и оптимизации работы ASIC-майнеров в реальном времени, которая позволяет отслеживать ключевые показатели оборудования и получать уведомления о любых проблемах. Сайт сервиса представлен на рисунке 3.

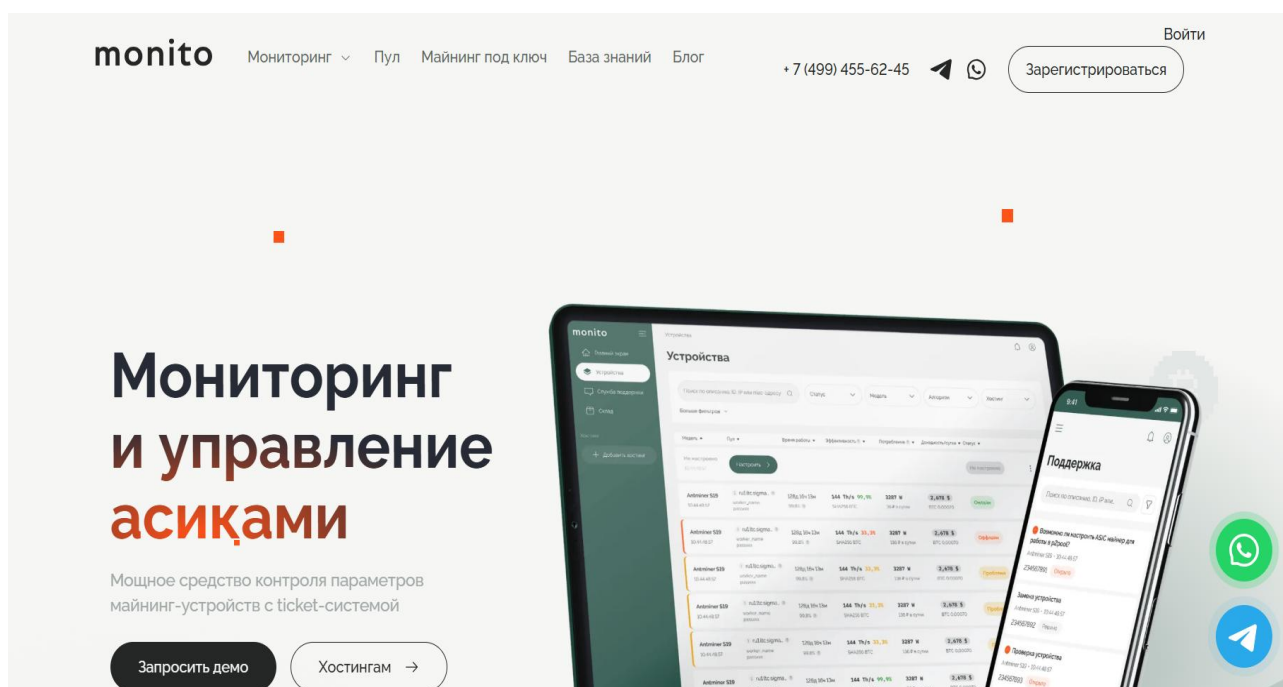


Рисунок 3 — Monito

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.18.001 ПЗ

Лист

12

Сравнения преимуществ и недостатков представлены в таблице 1.

Таблица 1 — Сравнительная таблица аналогов

Производитель	Преимущества	Недостатки
«Minerstat» (https://minerstat.com/)	Подходит и для GPU, и для ASIC, глубокая аналитика, автоматизация, инструменты для разгона	Сложность установки и настройки сервиса, отсутствие создания отчетности
Bixbit (https://bixbit.io/)	Не требует установки сторонних приложений, интуитивно-понятный интерфейс	Нет возможности сохранять данные в Excel-формате
Monito.run (https://monito.run/)	Есть возможность сохранения данных об устройствах, удобная настройка самих устройств	Сложность в установке агента для мониторинга за майнерами

На основании проведенного анализа был сделан вывод, что на рынке программ для составления отчетности майнинг—центра имеется потенциал для разработки новых решений.

1.3 Анализ требований к разработке

В рамках проекта разрабатывается программа для сбора, мониторинга и анализа данных с устройств для майнинга, зарегистрированных на платформе «Pitbit». Программа предназначена для автоматизации процесса сбора технических показателей, их визуализации и последующего экспорта в структурированные отчеты, что критически важно для эффективного управления парком оборудования в майнинг-центрах.

Требуется следующий функционал:

- вход в личный кабинет платформы;
- автоматический сбор и парсинг данных;
- отображение собранных данных в табличном виде;
- экспорт данных в Excel-таблицы;
- импорт данных из Excel-таблицы;
- интуитивно-понятный интерфейс;
- отображение подробной информации.

Помимо этого, для реализации и корректной работы программы были задействованы следующие программные компоненты и библиотеки:

- Selenium WebDriver, для автоматизации действий в браузере Chrome с возможностью открытия страницы, ввода логина/пароля, нажатия кнопок с целью получения динамически загружаемого контента;
- HtmlAgilityPack, для парсинга и навигации по полученному HTML-коду страницы с данными майнеров;
- ClosedXML, для работы с файлами Excel при импорте, экспорте и обновлении таблицы "hash-реестра";
- .NET WinForms, для создания графического интерфейса пользователя.

1.4 Выбор архитектуры системы

Для организации корректного и отказоустойчивого взаимодействия между пользователем, графическим интерфейсом и внешними веб-сервисами, работа приложения построена на архитектуре с асинхронными операциями. Основной рабочий цикл программы можно разбить на следующие этапы:

- ожидание и обработка пользовательского взаимодействия с элементами графического интерфейса. Это генерирует событие, которое перехватывается соответствующим обработчиком;

– в обработчике события инициируется асинхронная операция. Это позволяет не блокировать поток пользовательского интерфейса на время выполнения длительных задач;

– сбор и первичная обработка данных, если для веб-сбора: Selenium получает HTML-код целевой страницы. Затем этот код передается парсеру, который извлекает структурированные данные о каждом майнере, а если для файловых операций, тогда данные напрямую считываются из или записываются в Excel-файл;

– извлеченные или загруженные данные преобразуются в экземпляры внутреннего класса модели MinerData и сохраняются в главном списке, после чего в зависимости от текущего фильтра формируется список для отображения текущих устройств и вызывается метод обновления таблицы, который привязывает новые данные к элементу управления таблицы отображаемых устройств;

– после завершения всех операций обновляется строка состояния, счетчик устройств и при необходимости выводятся информационные диалоговые окна, информирующие пользователя о результате выполненного действия;

– программа вновь переходит в режим ожидания следующего пользовательского события, готовая к выполнению нового цикла. Время выполнения цикла зависит от сложности операции (скорости интернета, объема данных, производительности файловой системы).

Функциональная схема выполнения программы показана на рисунке 4.

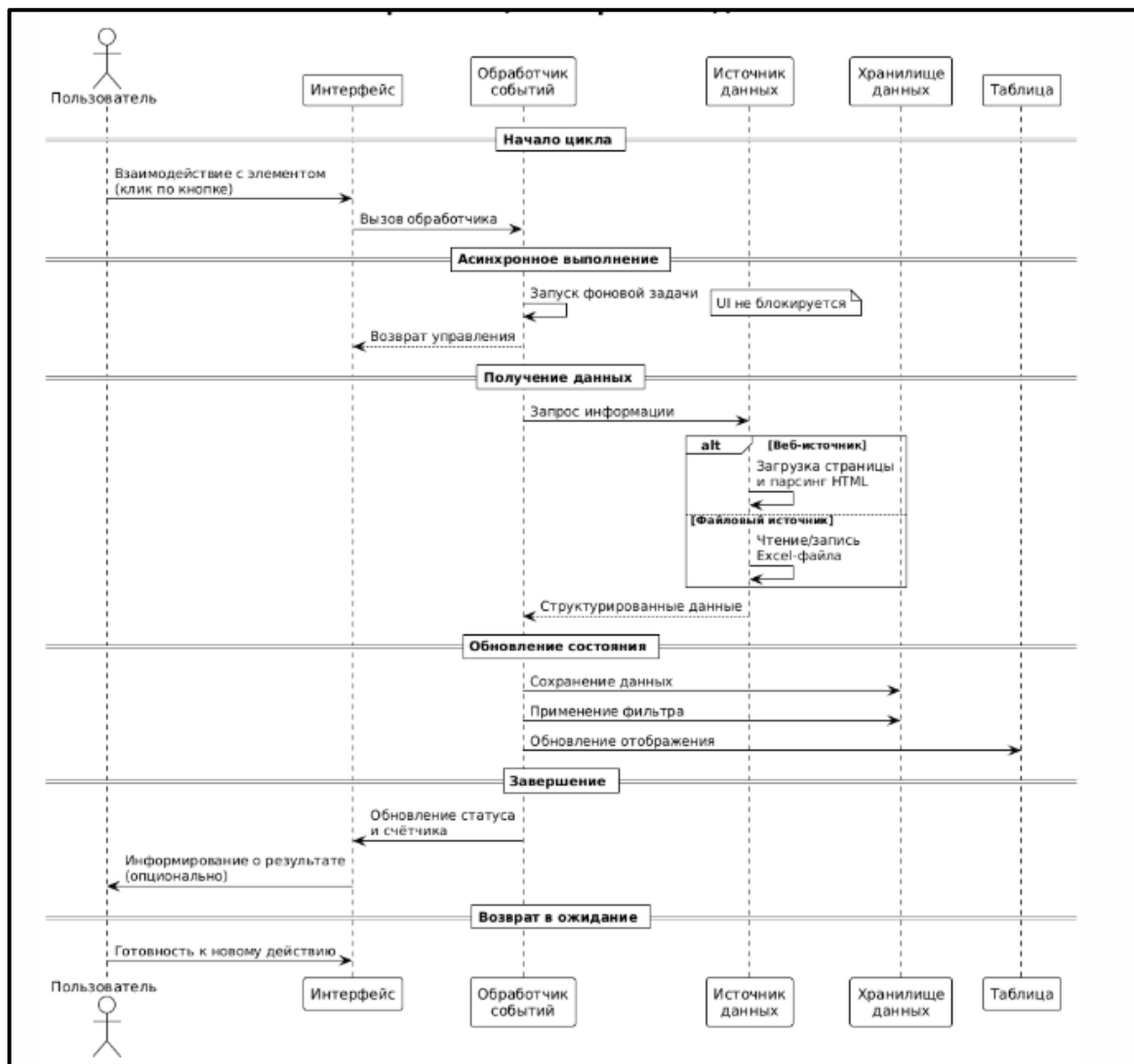


Рисунок 4 — Функциональная схема работы проекта

Приложение имитирует действия пользователя: переход на страницу авторизации, ввод учетных данных, навигацию к разделу с перечнем оборудования и извлечение необходимой информации из структуры веб-страницы. Детально процесс авторизации и последующего сбора данных показан на диаграммах последовательностей (рисунки 5 и 6).

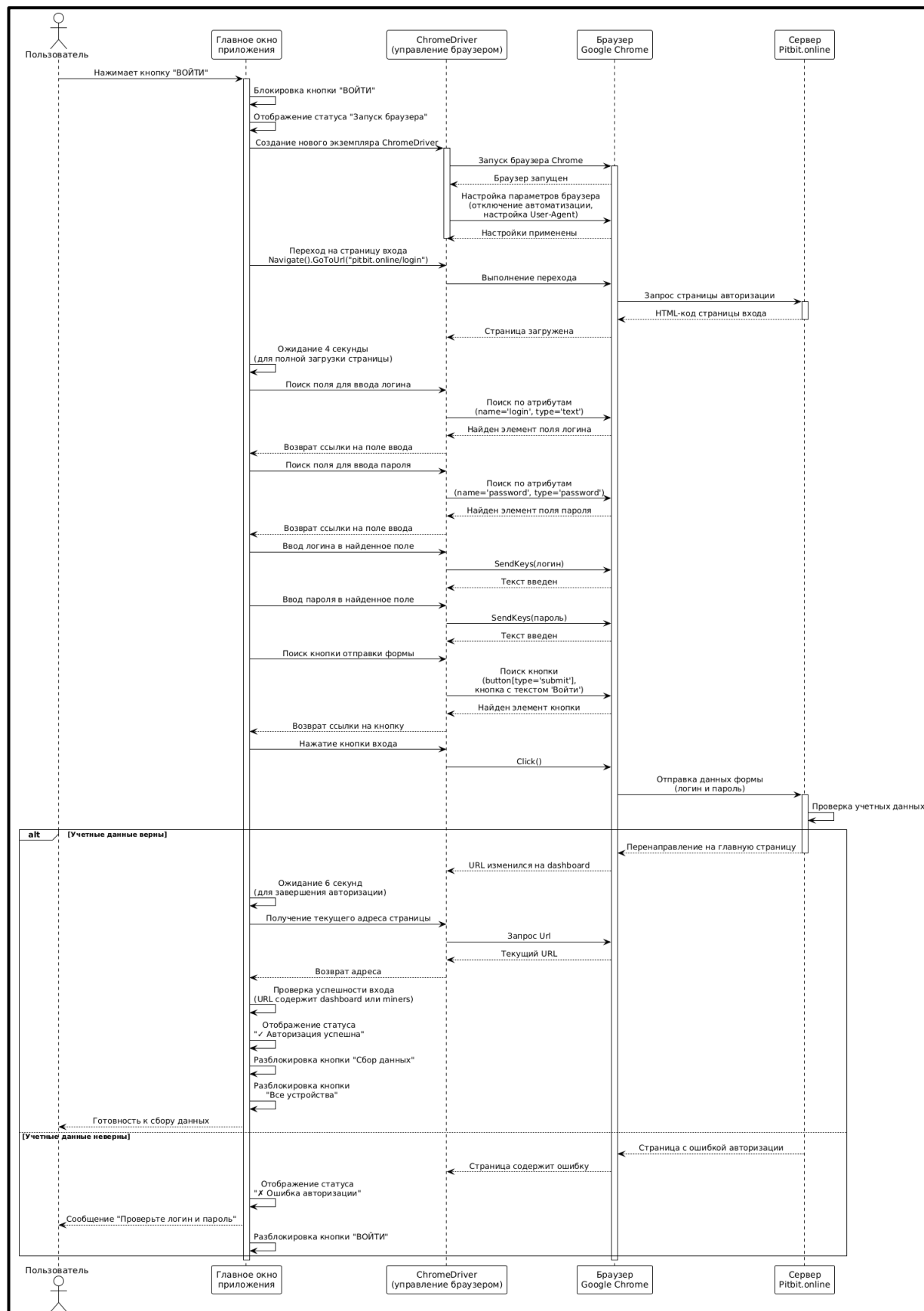


Рисунок 5 — Диаграмма последовательностей «Авторизация»

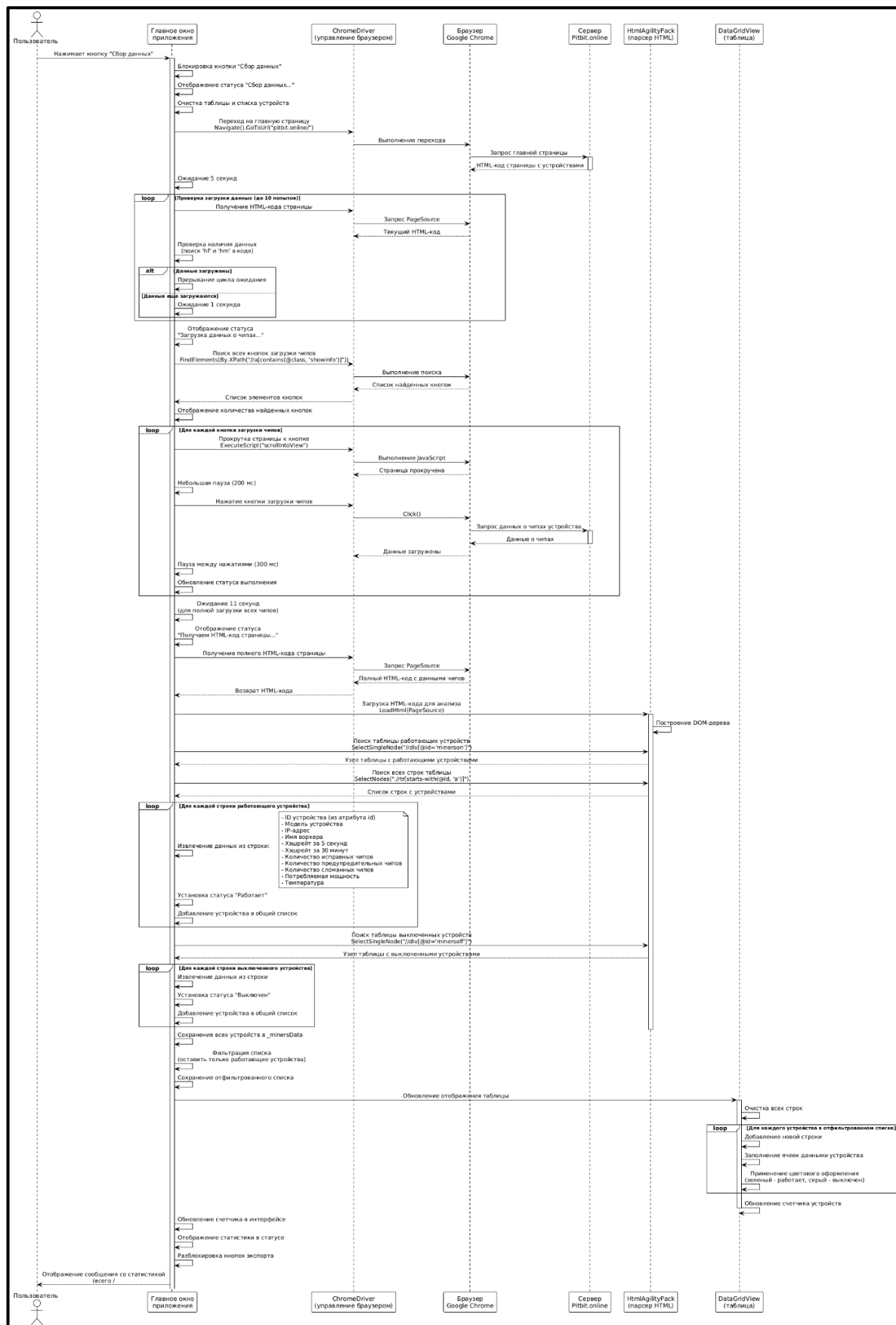


Рисунок 6 — Диаграмма последовательностей «Сбор данных»

После получения данных с веб-платформы приложение выполняет их обработку и представляет пользователю в табличном виде.

Развертывание приложения предполагает его установку на рабочую станцию пользователя с операционной системой Windows. Для корректной работы необходим установленный браузер Google Chrome, используемый в качестве среды для автоматизированного взаимодействия с веб-платформой Pitbit.Online. Программа взаимодействует с удаленным сервером по защищенному протоколу HTTPS и сохраняет результирующие файлы в локальной файловой системе. Архитектура развертывания системы отражена на рисунке 7.

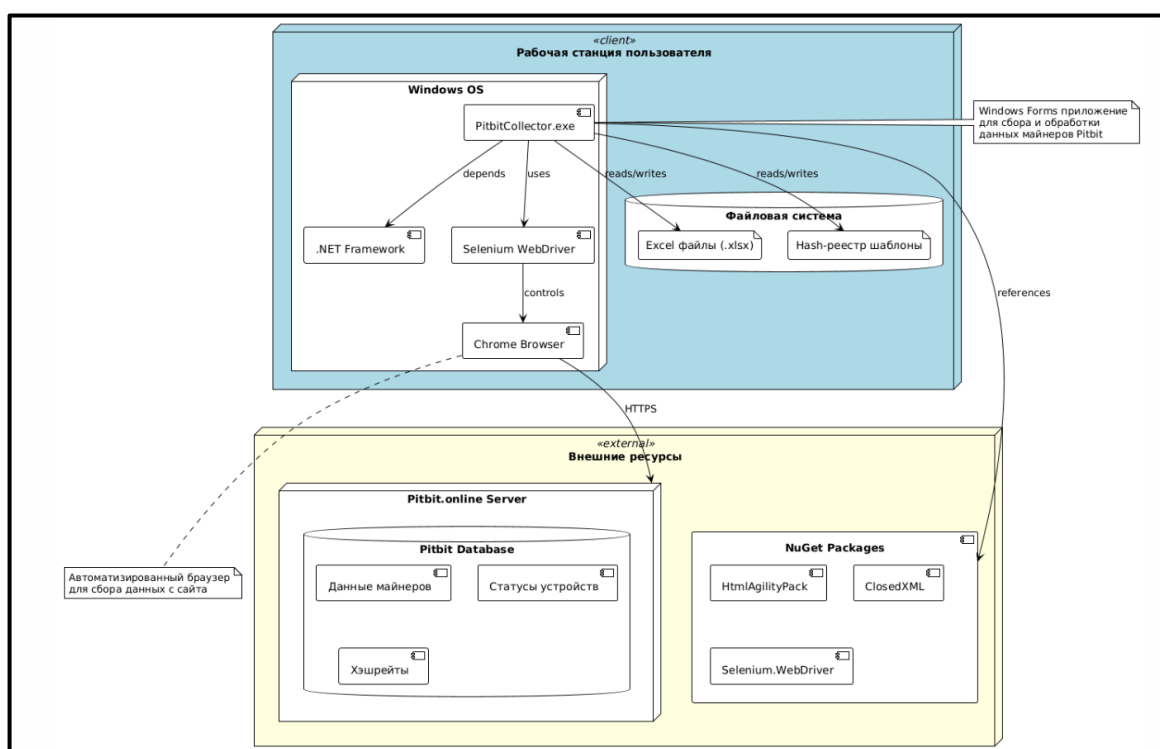


Рисунок 7 — Диаграмма развертывания

Исходя из требуемого функционала, планируемой архитектуры проекта и выбранной целевой платформы, был определен следующий набор инструментов разработки: язык программирования C#, платформа .NET Framework / .NET Core, среда разработки Microsoft Visual Studio 2022, операционная система Microsoft Windows 10/11.

1.5 Выбор языка программирования

C# (C Sharp) — объектно-ориентированный язык программирования с безопасной системой типов, разработанный компанией Microsoft. Предназначен для разработки приложений под платформу .NET, включая настольные приложения, веб-сервисы и облачные решения. Программа на C# представляет собой набор классов, содержащих методы и свойства, выполняемые последовательно в соответствии с логикой приложения. Классы могут наследоваться от базовых классов и реализовывать интерфейсы, что позволяет создавать гибкую и расширяемую архитектуру [2].

При программировании используются наборы библиотечных классов из .NET Framework / .NET Core, а также собственные классы, написанные на C#. Класс — это шаблон для создания объектов, объединяющий данные (поля, свойства) и поведение (методы, события). Встроенные библиотеки предоставляют широкий набор готовых компонентов: классы для работы с графическим интерфейсом, файловой системой, сетью, коллекциями и др.

Каждый отдельный метод представляет собой последовательность инструкций, выполняемых строго в порядке их написания. Методы могут вызывать другие методы, создавая иерархию вызовов. Внутри метода инструкции выполняются последовательно сверху вниз, за исключением случаев использования управляющих конструкций. Результат выполнения метода возвращается в вызывающий код либо сохраняется в полях класса для дальнейшего использования [2].

Ключевой особенностью C# является наличие событийно-ориентированной модели программирования, где действия пользователя генерируют события, которые обрабатываются специальными методами-обработчиками. Это позволяет создавать интерактивные приложения с графическим интерфейсом, реагирующие на действия пользователя в реальном времени.

1.6 Выбор среды разработки

В качестве среды разработки, исходя из целевой платформы проекта, была взята Microsoft Visual Studio 2022. Для создания графического интерфейса используется технология Windows Forms.

Visual Studio 2022 — интегрированная среда разработки (IDE) от компании Microsoft для создания программ на платформе .NET, а также для веб-разработки, мобильных приложений и облачных сервисов. Среда позволяет разрабатывать приложения для операционных систем семейства Windows с использованием языка программирования C# [3].

Visual Studio 2022 предоставляет широкий набор инструментов для разработки:

- визуальный конструктор форм позволяет создавать графический интерфейс пользователя путем перетаскивания элементов управления (кнопки, текстовые поля, таблицы, панели) из панели инструментов на рабочую область формы. Это существенно ускоряет разработку и позволяет сразу видеть, как будет выглядеть приложение [3];

- редактор кода с подсветкой синтаксиса обеспечивает удобное написание программного кода на C# с автоматическим дополнением (IntelliSense), подсказками по параметрам методов и быстрой навигацией по проекту;

- отладчик позволяет выполнять программу по шагам, устанавливать точки останова, отслеживать значения переменных в реальном времени, что необходимо для поиска и устранения ошибок [3];

- менеджер пакетов NuGet дает возможность легко подключать сторонние библиотеки, такие как Selenium WebDriver, HtmlAgilityPack и ClosedXML, без необходимости ручного скачивания и настройки.

Пример фрагмента программы в среде Visual Studio 2022 представлен на рисунке 8:

```

1 namespace primer
2 {
3     Ссылка: 0
4     internal static class Program
5     {
6         /// <summary>
7         /// The main entry point for the application.
8         /// </summary>
9         [STAThread]
10        Ссылка: 0
11        static void Main()
12        {
13            // To customize application configuration such as set high DPI settings or default font,
14            // see https://aka.ms/applicationconfiguration.
15            ApplicationConfiguration.Initialize();
16            Application.Run(new Form1());
17        }
18    }
19 }

```

Рисунок 8 — пример фрагмента алгоритма

Visual Studio 2022 проста в освоении для начинающих разработчиков благодаря интуитивно понятному интерфейсу и обширной документации. Визуальное проектирование форм позволяет быстро создавать прототипы приложений, а мощные средства отладки помогают находить и исправлять ошибки. Среда четко отражает связь между визуальными компонентами на форме и кодом, который обрабатывает события от этих компонентов. При условии качественной проработки архитектуры приложения с позиции разделения логики и представления, разработка в Visual Studio 2022 получается более эффективной, чем при использовании простых текстовых редакторов. Данная среда программирования бесплатна. Сайт Visual Studio представлен на рисунке 9.

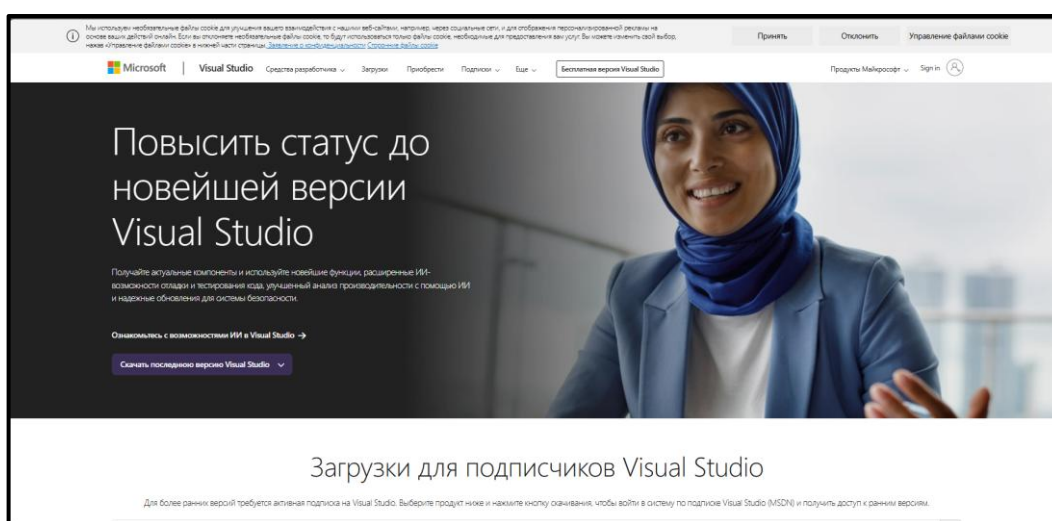


Рисунок 9 — Сайт Visual Studio

Страница загрузки Visual Studio включает:

- подробное описание функциональных возможностей среды;
- технические требования к системе;
- другие возможные версии;
- форму для обратной связи;
- кнопки загрузки установщика.

Для начала загрузки среды разработки необходимо нажать «Скачать» в области загрузки, что можно увидеть на рисунке 10.

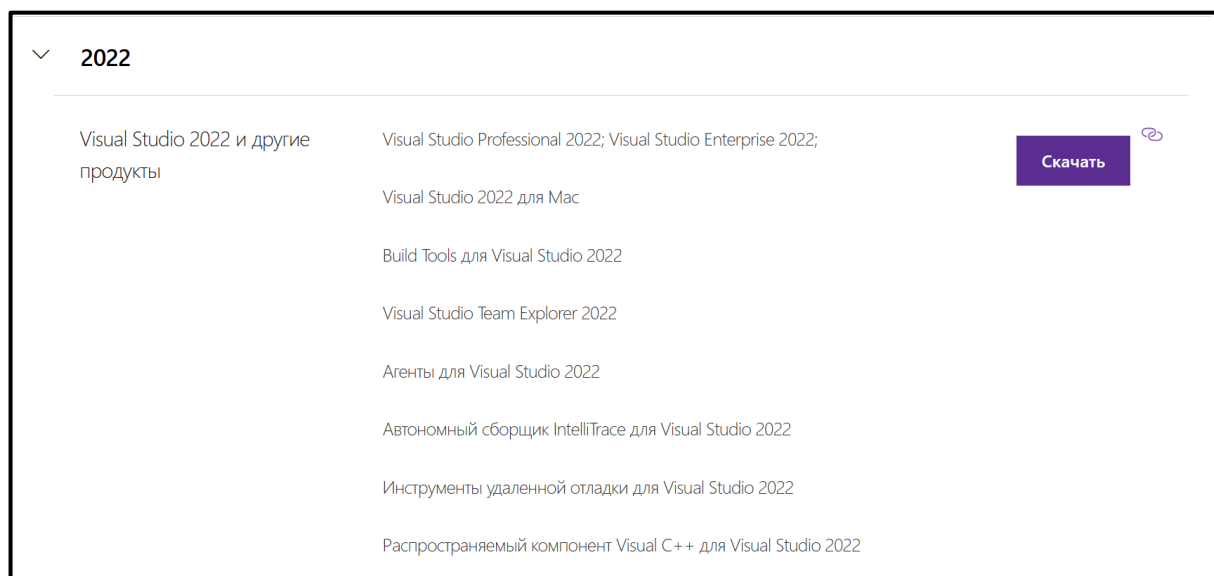


Рисунок 10 — Область загрузки среды программирования на сайте

После мы переходим на другую страницу для выбора версии функционала нашей среды разработки, где выбираем Visual Studio Professional 2022, и нажимаем скачать (данное окно показано на рисунке 11).

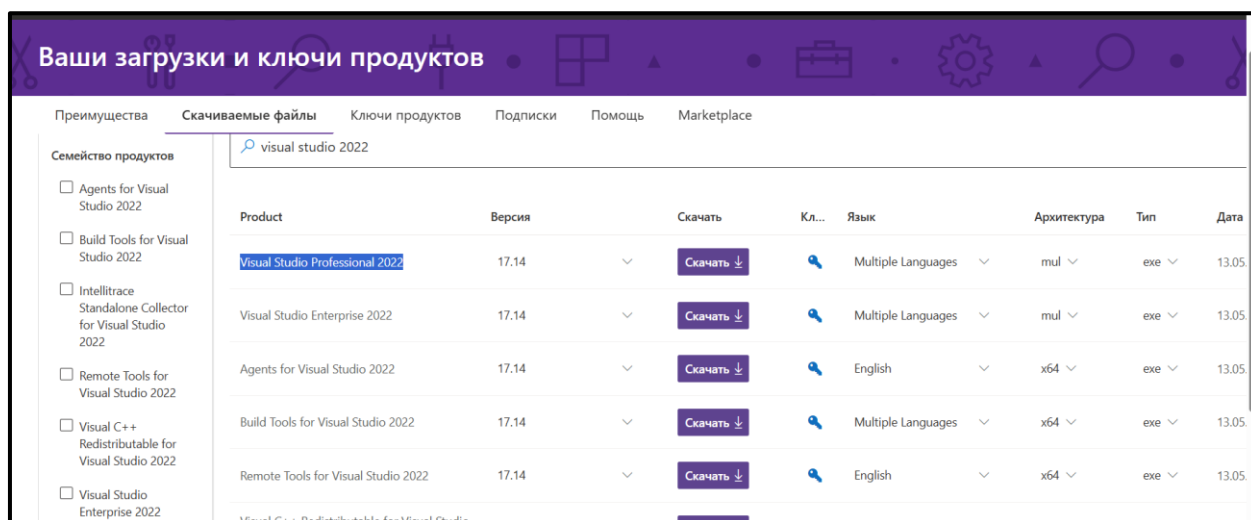


Рисунок 11 — Окно с выбором версий установщика

После нажатия кнопки "Скачать" запускается процесс загрузки исполняемого файла для установки среды программирования. Ход загрузки отображается в интерфейсе браузера и проиллюстрирован на рисунке 12.

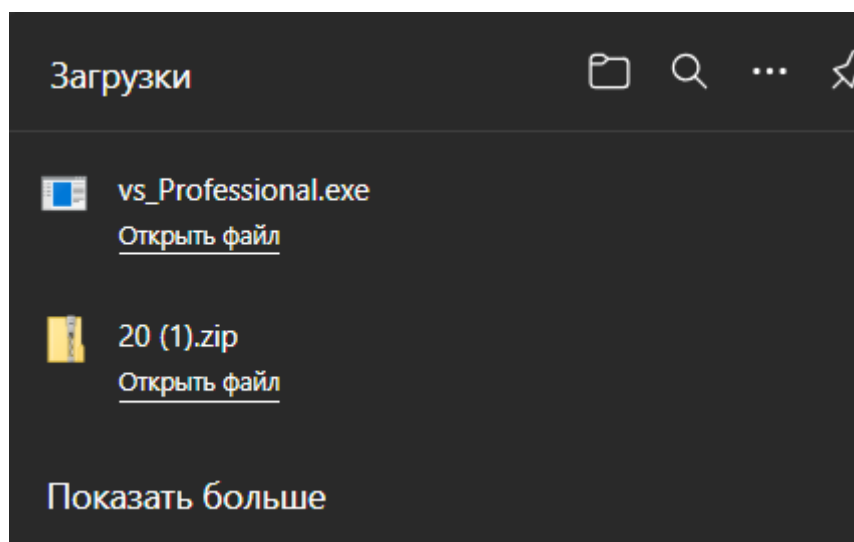


Рисунок 12 — Загрузка установочного файла среды программирования

После открываем установочный файл и ожидаем процесс установки скачанного ранее файла загрузчика. На рисунке 13 можно увидеть процесс его установки.

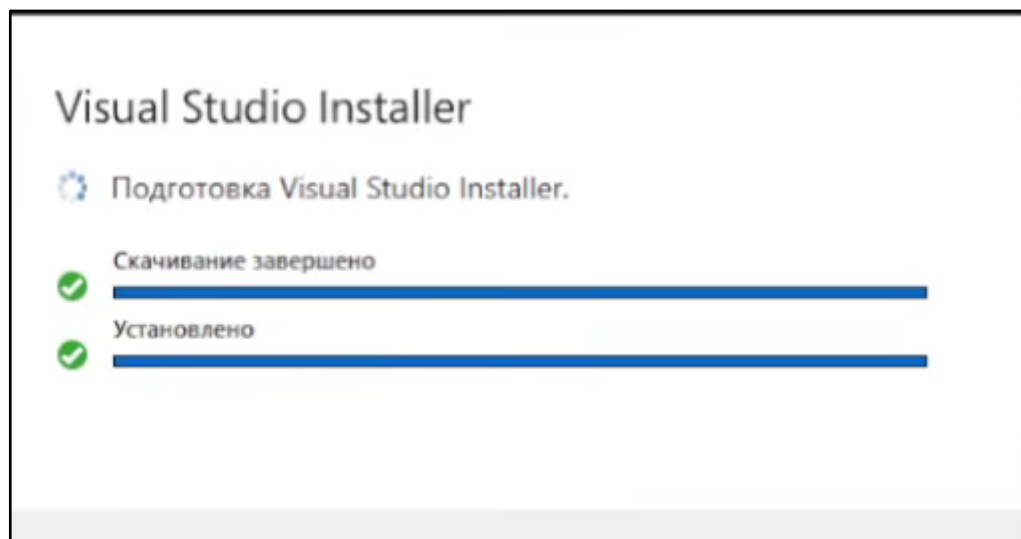


Рисунок 13 — Процесс установки

Затем появляется окно с выбором необходимых средств для разработки проектов. Необходимо выбрать «Разработка классических приложений .NET». И нажать на кнопку «Установить». Выбор рабочих нагрузок можно увидеть на рисунке 14.

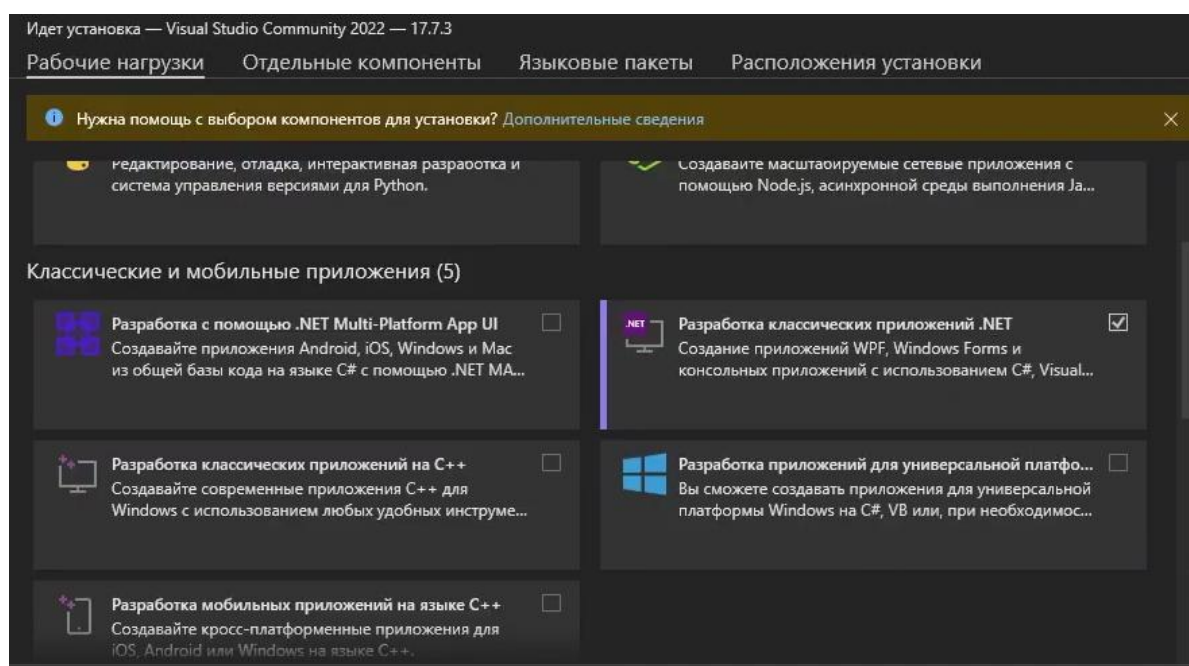


Рисунок 14 — Выбор параметров установки

После установки на главном экране можно увидеть установленную версию среды Microsoft Visual Studio 2022 (рисунок 15). Нажимаем на кнопку запустить и выполняем вход в меню среды, как показано на рисунке 16.

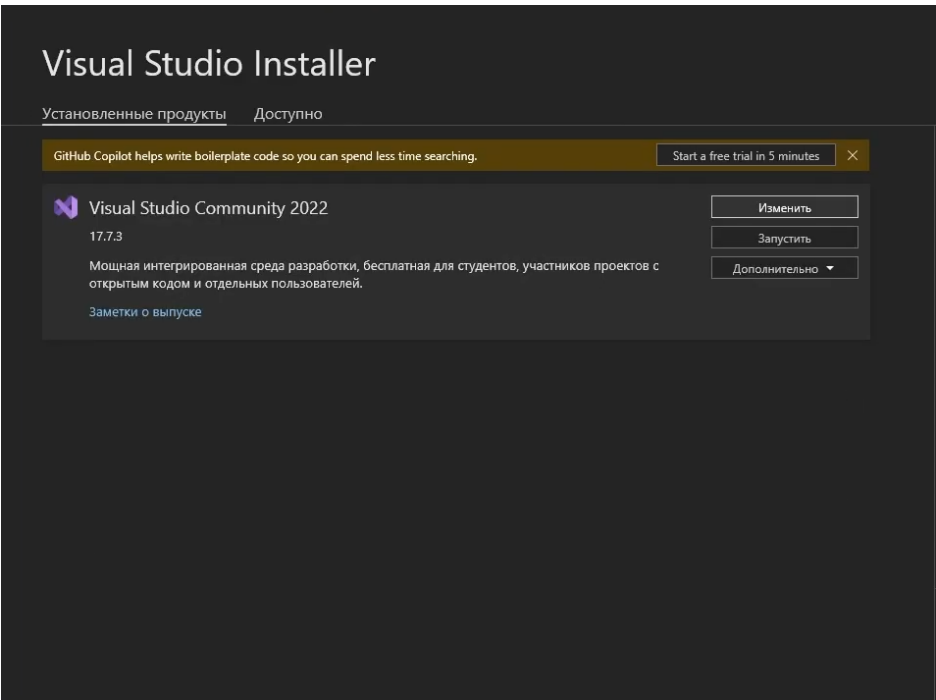


Рисунок 15 — Установленная версия среды разработки

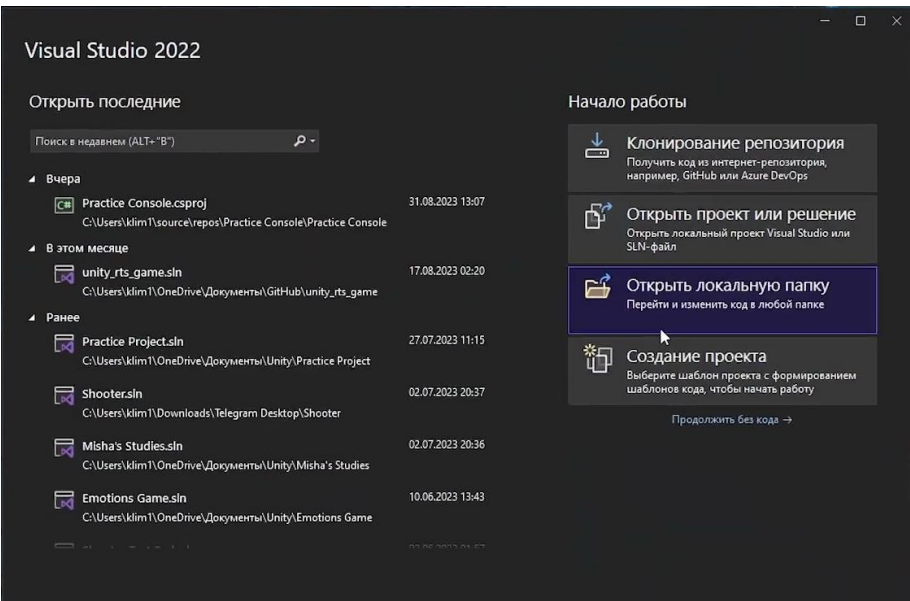


Рисунок 16 — Главное меню Visual Studio

Затем нажатием на кнопку «Создание проекта» переходим к настройкам разрабатываемого приложения. На рисунках 17 и 18 продемонстрирован выбор шаблона проекта и процесс настройки.

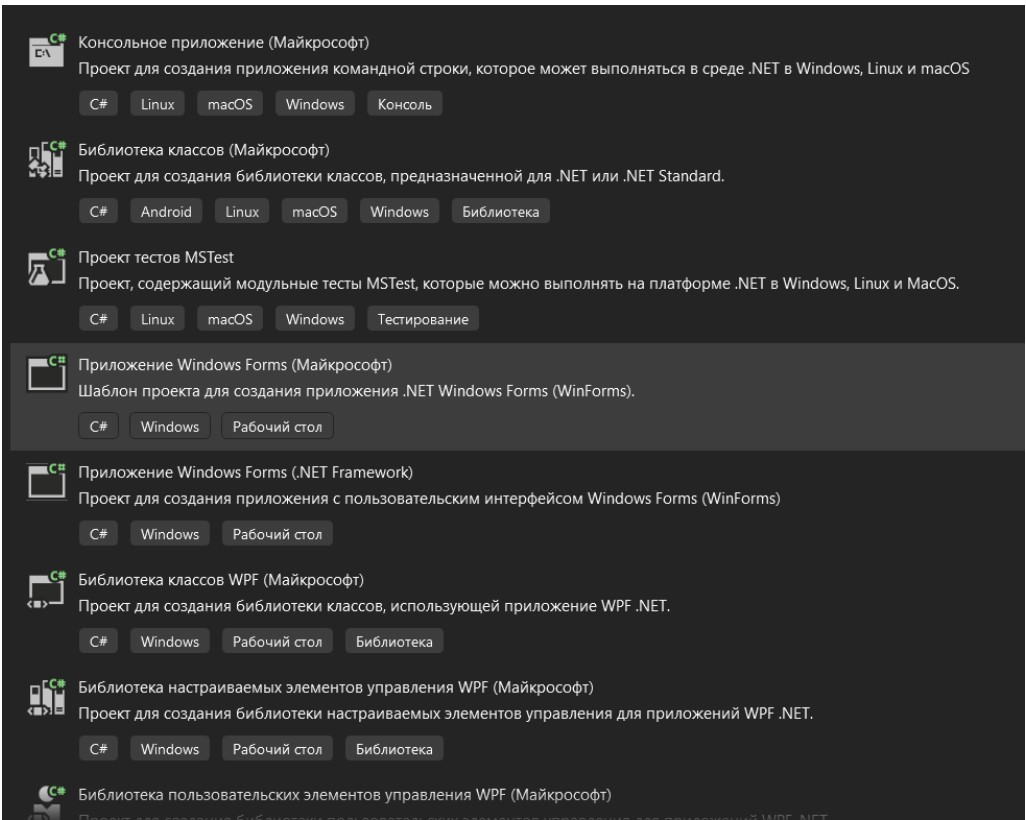


Рисунок 17 — Выбор шаблона проекта

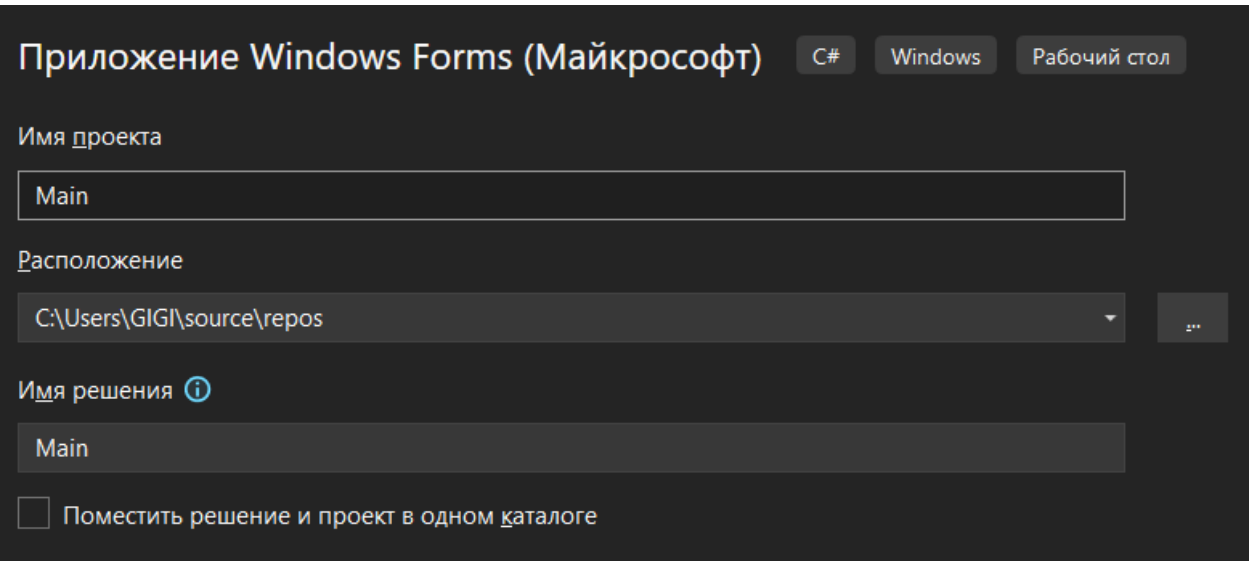


Рисунок 18 — Настройка среды разработки

Для реализации дипломного проекта были внедрены дополнительные библиотеки с помощью управления пакетами NuGet, сам процесс показан на рисунке 19, а необходимые библиотеки на рисунке 20.

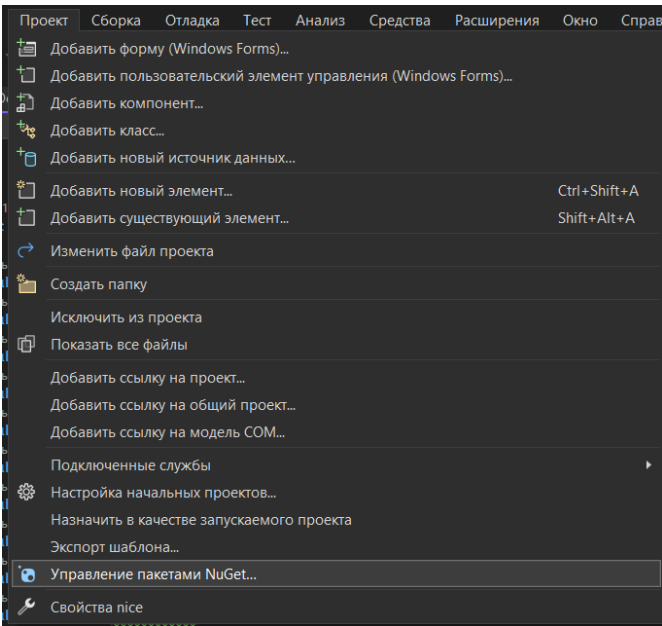


Рисунок 19 — Управление пакетами NuGet

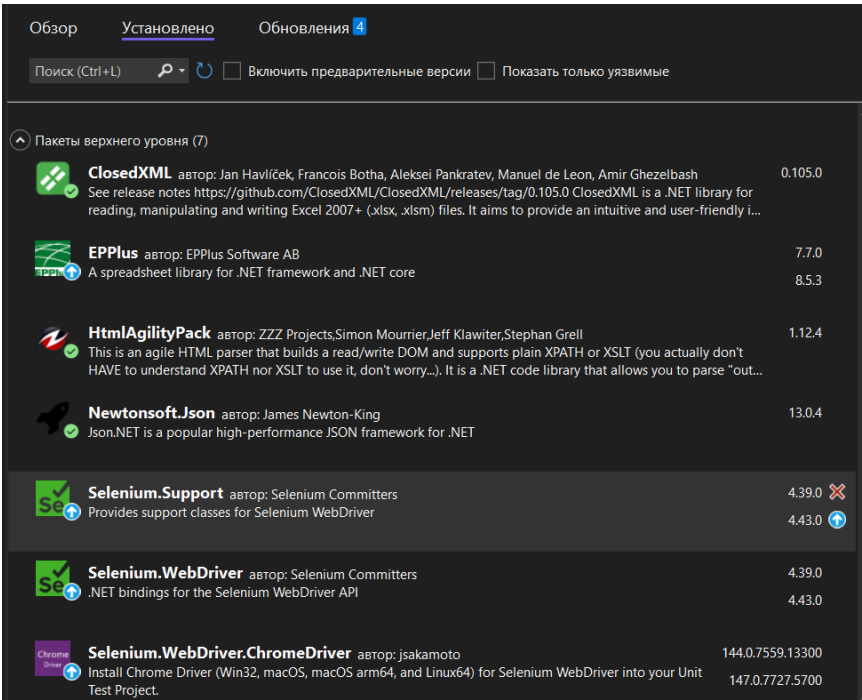


Рисунок 20 — Необходимые библиотеки в системе NuGet

1.7 Проектирование и разработка

Проект представляет собой приложение Pitbit Data Collector, разработанное для автоматизированного сбора и систематизации данных о майнинговом оборудовании с облачной платформы Pitbit.Online. Приложение создано в среде разработки с использованием языка C# и платформы Windows Forms, что обеспечивает привычный оконный интерфейс для пользователя и стабильную работу под управлением операционной системы Windows.

Приложение состоит из основного программного модуля, объединяющего графический интерфейс пользователя и логику обработки данных. Структурная организация программы отражена на диаграмме классов, представленной на рисунке 21.

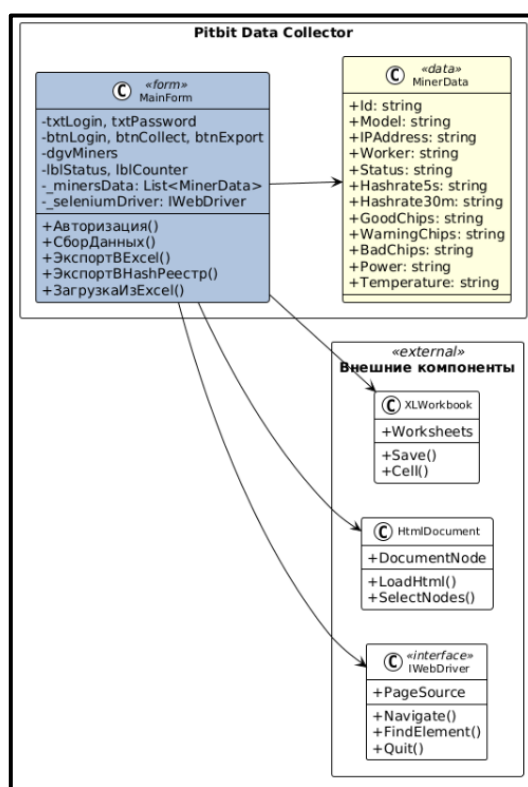


Рисунок 21 — Диаграмма классов

Графический интерфейс приложения включает панель авторизации с полями ввода логина и пароля, панель управления с кнопками выполнения основных операций, а также табличную область для отображения собранных

данных об устройствах. В таблице выводятся следующие параметры майнеров: идентификатор, модель, IP-адрес, наименование воркера, значения хэшрейта за 5 секунд и 30 минут, количество исправных, предупредительных и сломанных чипов, потребляемая мощность, температура и текущий статус устройства.

Пользователь взаимодействует с программой через набор управляющих элементов, инициируя такие операции, как авторизация в системе Pitbit, сбор актуальных данных об оборудовании, переключение между режимами отображения устройств, экспорт информации в формат Excel, формирование специализированного hash-реестра, а также загрузка ранее сохраненных данных.

Автоматизация сбора данных реализована с применением средств веб-автоматизации, позволяющих программно управлять браузером.

После получения данных с веб-платформы приложение выполняет их обработку и представляет пользователю в табличном виде. Предусмотрена цветовая индикация строк в зависимости от статуса устройства и количества исправных чипов, что упрощает визуальный анализ состояния оборудования. Пользователь имеет возможность просмотреть детальную информацию по любому устройству двойным щелчком по соответствующей строке таблицы.

Экспорт данных реализован в двух форматах. При стандартном экспорте в Excel создается файл с полным перечнем отображаемых устройств и сохранением цветового форматирования строк. При экспорте в hash-реестр данные распределяются по заданным диапазонам ячеек в соответствии с наименованиями воркеров и сортируются по убыванию производительности. Последовательность операций при экспорте данных иллюстрируется соответствующей диаграммой (рисунок 22).

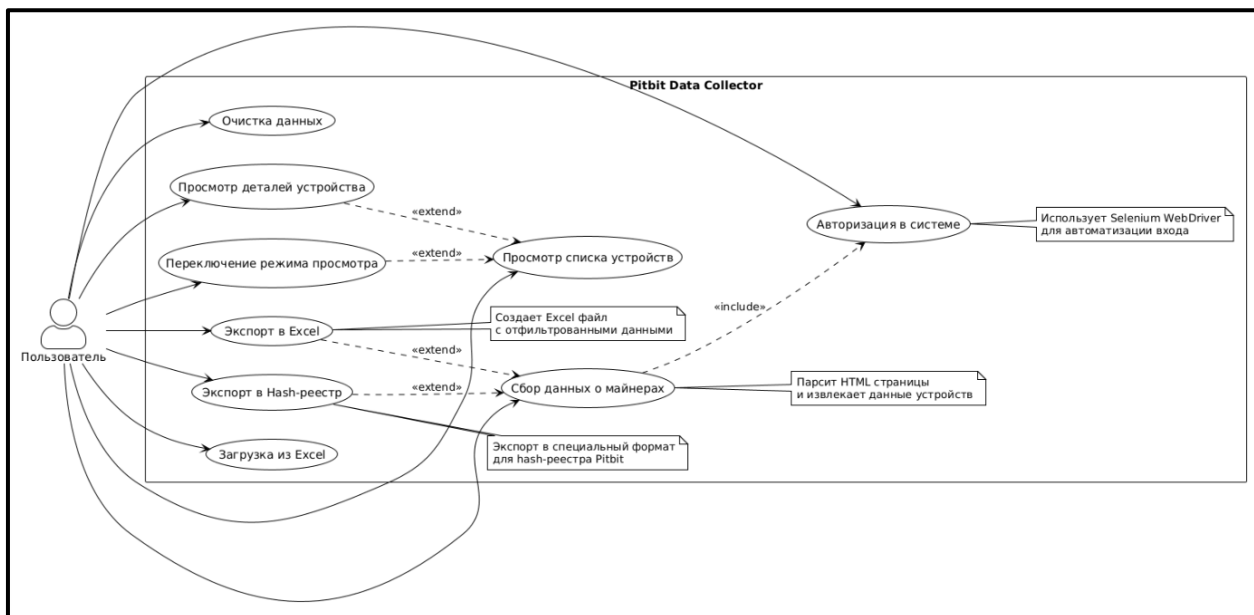


Рисунок 22 — Диаграмма прецедентов пользователя

Для удобства повторного анализа данных реализована функция загрузки информации из ранее сохраненных Excel-файлов, что исключает необходимость повторного обращения к веб-интерфейсу платформы.

Дипломный проект автоматизирует процессы мониторинга состояния майнингового оборудования и подготовки отчетной документации, что способствует повышению эффективности администрирования оборудования.

1.8 Разработка управляющей программы

Проект представляет собой приложение, разработанное для автоматизированного сбора и систематизации данных о майнинговом оборудовании с облачной платформы Pitbit.Online. Приложение создано на языке C# с использованием платформы Windows Forms, что обеспечивает привычный оконный интерфейс и стабильную работу под управлением операционной системы Windows.

Структурная организация программы отражена на диаграмме классов. Центральным элементом архитектуры является класс MainForm, который наследуется от базового класса Form и представляет главное окно

приложения. Данный класс инкапсулирует все элементы пользовательского интерфейса, логику обработки событий и координацию работы с внешними компонентами. Модель данных представлена классом MinerData, экземпляры которого используются для хранения информации об отдельных устройствах.

1.8.1 Инициализация пользовательского интерфейса

При запуске приложения вызывается конструктор класса MainForm, который выполняет инициализацию компонентов графического интерфейса посредством вызова метода InitializeComponents(). В данном методе программно создаются и настраиваются все элементы управления, что исключает необходимость использования дизайнера форм и обеспечивает полный контроль над процессом создания интерфейса.

Форма содержит две основные панели. Панель авторизации panel1 включает текстовые поля для ввода логина и пароля, а также кнопку «ВОЙТИ», запускающую процесс авторизации на платформе Pitbit. Панель действий panel2 содержит набор кнопок для выполнения основных операций: сбор данных, загрузка из Excel, переключение режима отображения, экспорт в Excel, формирование hash-реестра и очистка данных. В нижней части панели расположены метки для отображения счетчика устройств и текущего статуса программы.

Центральное место в интерфейсе занимает таблица DataGridView с идентификатором dgvMiners, предназначенная для отображения собранных данных. В листинге 1 представлен фрагмент кода, отвечающий за настройку столбцов таблицы.

Листинг 1 — Настройка столбцов таблицы данных

```
dgvMiners.Columns.Add("ID", "ID");  
dgvMiners.Columns.Add("Модель", "Модель");  
dgvMiners.Columns.Add("IP", "IP Адрес");  
dgvMiners.Columns.Add("Воркер", "Воркер");  
dgvMiners.Columns.Add("Хэшрейт 5с", "Хэшрейт 5с (TH/s)");  
dgvMiners.Columns.Add("Хэшрейт 30мин", "Хэшрейт 30мин (TH/s)");  
dgvMiners.Columns.Add("Чипы (хорошие)", "Чипы (хорошие)");
```

```

dgvMiners.Columns.Add("Чипы (плохие)", "Чипы (плохие)");
dgvMiners.Columns.Add("Чипы (слом.)", "Чипы (слом.)");
dgvMiners.Columns.Add("Всего чипов", "Всего чипов");
dgvMiners.Columns.Add("Мощность", "Мощность (W)");
dgvMiners.Columns.Add("Температура", "Температура (°C)");
dgvMiners.Columns.Add("Статус", "Статус");

```

Для улучшения визуального восприятия данных в таблице реализован обработчик события `CellFormatting`, который применяет цветовое оформление строк в зависимости от статуса устройства. Работающие устройства выделяются зеленым цветом, выключенные — серым, скрытые — светло-красным. Также цветом выделяются ячейки с количеством чипов: исправные чипы отображаются на зеленом фоне, предупредительные — на желтом, сломанные — на красном.

1.8.2 Авторизация на платформе Pitbit

Процесс авторизации реализован в асинхронном методе `LoginToPitbitSelenium()` и использует библиотеку `Selenium WebDriver` для автоматизированного управления браузером `Google Chrome`. При нажатии кнопки «ВОЙТИ» создается экземпляр драйвера `Chrome` с предварительно настроенными параметрами, отключающими обнаружение автоматизации со стороны веб-сайта. Фрагмент настройки драйвера представлен в листинге 2.

Листинг 2 — Настройка `ChromeDriver`

```

var options = new ChromeOptions();
options.AddArgument("--disable-blink-features=AutomationControlled");
options.AddExcludedArgument("enable-automation");
options.AddAdditionalOption("useAutomationExtension", false);
options.AddArgument("user-agent=Mozilla/5.0...");
_seleniumDriver = new ChromeDriver(options);

```

После запуска браузера драйвер выполняет переход на страницу авторизации `https://pitbit.online/login`. С помощью методов `FindElements` осуществляется поиск полей ввода логина и пароля по различным селекторам, что повышает устойчивость к возможным изменениям структуры страницы. Найденные поля заполняются значениями из соответствующих текстовых

полей формы, после чего выполняется поиск и нажатие кнопки отправки формы.

Успешность авторизации определяется путем анализа текущего URL-адреса и содержимого страницы. При обнаружении ключевых слов, характерных для страницы после входа, приложение активирует кнопки сбора данных и переключения режима отображения, а также отображает соответствующее сообщение в строке статуса. Последовательность операций авторизации наглядно представлена на диаграмме последовательностей.

1.8.3 Сбор данных об устройствах

Основной функционал приложения по сбору данных реализован в асинхронном методе `CollectMinerDataSelenium()`. Процесс сбора данных начинается с перехода на главную страницу платформы Pitbit, после чего приложение ожидает загрузки динамического контента. Для определения момента загрузки данных выполняется циклическая проверка наличия в HTML-коде страницы характерных идентификаторов `hf` и `hm`, соответствующих элементам отображения хэшей.

Особенностью платформы Pitbit является то, что данные о состоянии чипов загружаются асинхронно только после нажатия специальных кнопок. Для решения этой задачи приложение выполняет поиск всех элементов с классом `showinfo`, содержащих атрибут `addi` со значением `chips`, и программно имитирует нажатие каждой такой кнопки. Фрагмент кода, выполняющий данную операцию, приведен в листинге 3.

Листинг 3 — Активация загрузки данных о чипах

```
var chipButtons = _seleniumDriver.FindElements(  
    By.XPath("//a[contains(@class, 'showinfo') and contains(@addi,  
'chips')]"));  
foreach (var button in chipButtons)  
    {((IJavaScriptExecutor)_seleniumDriver).ExecuteScript("arguments[0].scrollIntoView(true);", button);  
    button.Click();
```

```
await Task.Delay(300);
}
```

После нажатия всех кнопок приложение выдерживает паузу продолжительностью 11 секунд, необходимую для полной загрузки данных о чипах со стороны сервера. Затем выполняется получение полного HTML-кода страницы через свойство PageSource драйвера.

Для извлечения структурированных данных из HTML-кода используется библиотека HtmlAgilityPack. Полученный код загружается в объект HtmlDocument, после чего выполняется поиск таблиц с работающими и выключенными устройствами по идентификаторам minerson и minersoff соответственно. Для каждой найденной строки таблицы вызывается метод ParseMinerData(), который извлекает значения отдельных полей.

Метод ParseMinerData() осуществляет разбор строки таблицы, извлекая следующие данные: идентификатор устройства из атрибута id строки, модель устройства из ссылки с классом tdv, IP-адрес из ссылки, содержащей шаблон 192.168., имя воркера из ячейки с классом pools, значения хэшрейта из элементов с идентификаторами hf{id} и hm{id}, количество чипов различного состояния из соответствующих элементов, мощность и температуру. Для извлечения данных о чипах используются регулярные выражения, позволяющие найти значения в сложной структуре HTML-кода.

Собранные данные о каждом устройстве сохраняются в экземпляре класса MinerData, после чего добавляются в общий список _minersData. Статус устройства определяется в зависимости от того, в какой таблице была найдена строка: «Работает» для таблицы minerson, «Выключен» для таблицы minersoff.

1.8.4 Отображение и фильтрация данных

После завершения сбора данных приложение формирует отфильтрованный список устройств для отображения в таблице. По умолчанию активирован режим отображения только работающих устройств, что реализуется с помощью LINQ-запроса _minersData.Where(m => m.Status

== "Работает").ToList(). Отфильтрованный список сохраняется в переменной `_currentDisplayedMiners`.

Метод `UpdateDataGridView()` выполняет очистку таблицы и построчное заполнение данными из переданного списка. Для каждой строки устанавливаются значения всех столбцов в соответствии с полями объекта `MinerData`. После заполнения таблицы обновляется счетчик устройств в метке `lblCounter` и выводится статистическая информация в строку статуса.

Пользователь имеет возможность переключаться между режимами отображения с помощью кнопки, обработчик которой меняет значение флага `_showingOnlineOnly` и соответствующим образом обновляет список `_currentDisplayedMiners`. Текст кнопки изменяется на «Только работающие» или «Все устройства» в зависимости от текущего режима.

Для просмотра детальной информации об устройстве реализован обработчик двойного щелчка по строке таблицы `dgvMiners_CellDoubleClick`. При его срабатывании формируется текстовое описание выбранного устройства, содержащее все доступные параметры, и отображается в диалоговом окне.

1.8.5 Экспорт данных в Excel

Функция экспорта данных в формат Excel реализована в методе `ExportToExcel()` с использованием библиотеки `ClosedXML`. При вызове метода отображается диалоговое окно сохранения файла, в котором автоматически формируется имя файла, включающее тип отображаемых устройств и текущую дату со временем.

После выбора пути сохранения создается новый экземпляр рабочей книги `XLWorkbook` и добавляется лист с названием «Майнеры». Первая строка листа заполняется заголовками столбцов, для которых применяется жирное начертание и серый фон заливки. Затем в цикле выполняется построчная запись данных из списка `_currentDisplayedMiners`. Фрагмент кода записи данных представлен в листинге 4.

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

Листинг 4 — Запись данных в Excel

```
for (int i = 0; i < _currentDisplayedMiners.Count; i++)  
{  
    var miner = _currentDisplayedMiners[i];  
    int row = i + 2;  
    worksheet.Cell(row, 1).Value = miner.Id;  
    worksheet.Cell(row, 2).Value = miner.Model;  
    worksheet.Cell(row, 3).Value = miner.IPAddress;  
    if (miner.Status == "Работает")  
        worksheet.Row(row).Style.Fill.BackgroundColor = XLColor.LightGreen;  
    else if (miner.Status == "Выключен")  
        worksheet.Row(row).Style.Fill.BackgroundColor = XLColor.LightGray;  
}
```

Для каждой строки применяется цветовое оформление, соответствующее статусу устройства, что делает экспортированный файл наглядным и удобным для анализа. После заполнения всех данных вызывается метод `AdjustToContents()` для автоматической подгонки ширины столбцов под содержимое, и выполняется сохранение файла.

1.8.6 Экспорт в формат hash-реестра

Специализированный экспорт в формат hash-реестра реализован в методах `ExportToHashRegistry()` и `ExportToHashRegistryWorkbook()`. Данная функция предназначена для заполнения заранее подготовленного шаблона Excel в соответствии с внутренними правилами распределения устройств платформы Pitbit.

При вызове функции отображается диалог выбора файла шаблона, после чего пользователю предлагается выбрать лист из числа доступных в книге. Выбранный лист загружается, и выполняется очистка диапазонов ячеек, предназначенных для размещения данных. Очистка производится для двух диапазонов: столбцы B-F со строки 8 по 219 и столбцы H-L со строки 8 по 52.

Устройства распределяются по диапазонам в зависимости от имени воркера. Устройства с префиксами `workerlesbx` и `workerlesax` направляются в первый диапазон (B-F), а устройства с префиксом `workerles00x` — во второй

диапазон (H-L). Остальные устройства игнорируются. Внутри каждого диапазона устройства сортируются по убыванию значения хэшрейта за 30 минут.

Для каждого устройства в соответствующий диапазон записываются IP-адрес, модель, значения хэшрейта (преобразованные в целочисленный формат путем умножения на 1000) и количество исправных чипов. После заполнения данных выполняется обновление статистической информации о количестве устройств различных моделей с разбивкой по количеству исправных чипов. Статистика записывается в ячейки I2, J2, K2 (для моделей S19 Pro), O2, P2 (для S19 EXP) и S2, T2 (для S21 Hydro).

1.8.7 Загрузка данных из Excel

Функция загрузки ранее сохраненных данных реализована в методе LoadFromExcel(). При вызове метода отображается диалог выбора файла Excel, после подтверждения выбора открывается рабочая книга и выполняется чтение первого листа.

Чтение данных производится построчно, начиная со второй строки (первая строка содержит заголовки). Цикл продолжается до тех пор, пока первая ячейка текущей строки не окажется пустой. Для каждой строки создается новый экземпляр класса MinerData, поля которого заполняются значениями из соответствующих столбцов. Листинг 5 демонстрирует процесс чтения данных.

Листинг 5 — Чтение данных из Excel

```
int row = 2;
while (!worksheet.Cell(row, 1).IsEmpty())
{
    var miner = new MinerData
    {
        Id = worksheet.Cell(row, 1).GetValue<string>(),
        Model = worksheet.Cell(row, 2).GetValue<string>(),
        IPAddress = worksheet.Cell(row, 3).GetValue<string>(),
```

```
};
_minersData.Add(miner);
row++;
}
```

После завершения чтения данные загружаются в список `_minersData`, формируется отфильтрованный список работающих устройств и выполняется обновление таблицы. Пользователю отображается сводная информация о количестве загруженных устройств с разбивкой по статусам.

1.8.8 Вспомогательные функции и обработка событий

Приложение включает ряд вспомогательных функций, обеспечивающих корректную работу интерфейса и обработку данных. Обработчик изменения размеров формы `MainForm_Resize` динамически пересчитывает положение и размеры элементов управления при изменении размеров окна, что обеспечивает адаптивность интерфейса. Кнопка «Очистить» сбрасывает все внутренние списки данных, очищает таблицу и возвращает интерфейс в исходное состояние.

Для парсинга числовых значений из текстовых данных реализован метод `ParseChipValue()`, который удаляет все нецифровые символы из строки. Метод `ExtractTemperature()` извлекает значение температуры из HTML-элемента, обрабатывая как содержимое самого элемента, так и соседние текстовые узлы. Метод `CleanHtmlText()` выполняет очистку текста от HTML-тегов и лишних пробельных символов.

При закрытии формы вызывается переопределенный метод `OnFormClosing()`, который корректно завершает работу драйвера Selenium, освобождая системные ресурсы и закрывая окно браузера.

1.8.9 Архитектура развертывания

Развертывание приложения предполагает его функционирование на рабочей станции пользователя под управлением операционной системы Windows. Для работы приложения требуется установленный браузер Google Chrome и соответствующий драйвер ChromeDriver. Приложение

взаимодействует по защищенному протоколу HTTPS с удаленным сервером Pitbit.online, а также осуществляет чтение и запись файлов Excel в локальной файловой системе.

Таким образом, разработанное приложение Pitbit Data Collector представляет собой законченное программное решение, автоматизирующее процессы сбора, обработки, анализа и экспорта данных о майнинговом оборудовании с облачной платформы. Приложение обладает интуитивно понятным графическим интерфейсом, устойчиво к изменениям структуры веб-страниц и предоставляет гибкие возможности по работе с данными в формате Excel.

1.9 Графический интерфейс

В данном разделе будут описаны сценарии взаимодействия с графическим интерфейсом программы. Для общего понимания функциональности разработанной программы рассмотрим главный экран приложения на рисунке 23.

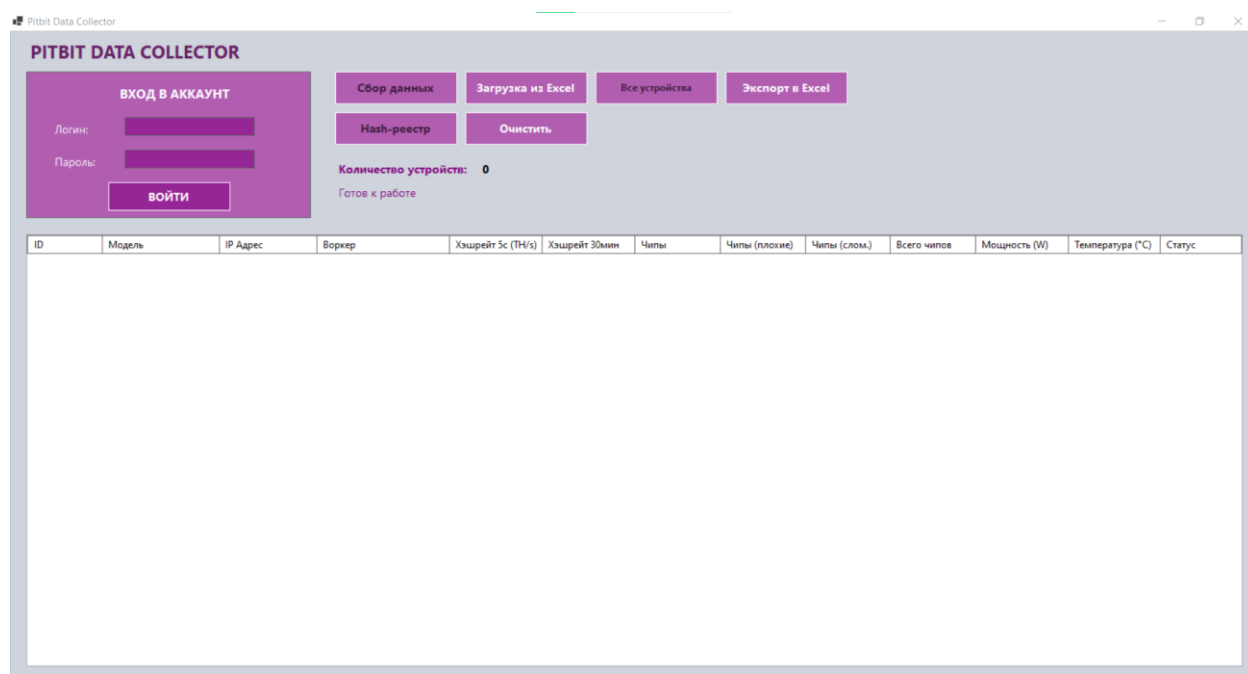


Рисунок 23 — Главный экран программы

Сразу после запуска пользователю доступны все возможности приложения. Для удобства работы экран разделен на несколько функциональных областей: панель авторизации с полями ввода логина и пароля, панель управления с кнопками основных операций, таблица для отображения данных об устройствах, а также строка состояния с информацией о количестве устройств и текущем статусе программы.

1.9.1 Сценарий «Авторизация в системе»

Для начала работы с программой пользователю необходимо выполнить вход в облачную платформу Pitbit. В левой части главного экрана расположена панель авторизации, содержащая поля для ввода логина и пароля, а также кнопку «ВОЙТИ». Пользователь вводит учетные данные и нажимает кнопку входа, как показано на рисунке 24.

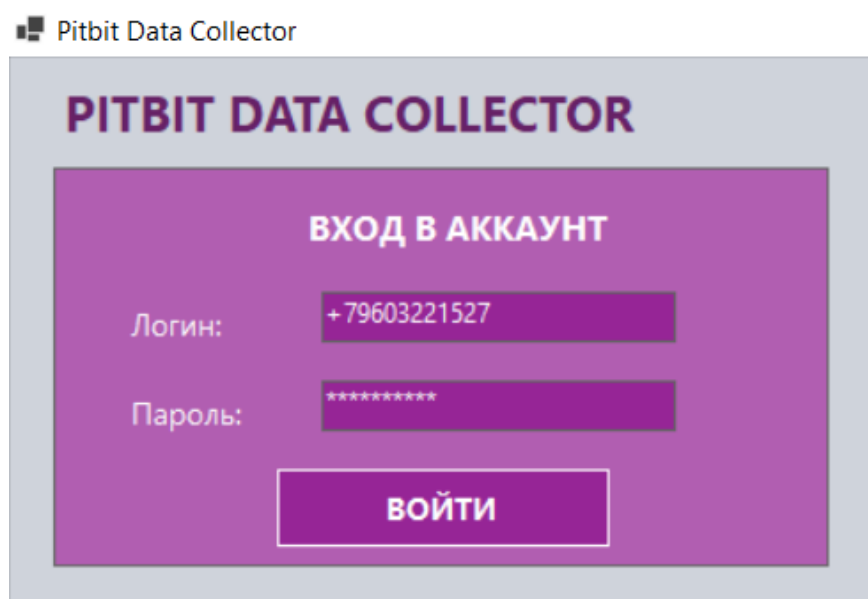


Рисунок 24 — Ввод учетных данных

После нажатия кнопки запускается процесс автоматизированной авторизации с использованием браузера Chrome. Программа отображает текущий статус выполнения операции в строке состояния: «Запуск браузера...», затем выполняется переход на страницу входа платформы Pitbit, заполнение полей формы и отправка данных. При успешной авторизации

строка состояния отображает сообщение «✓ Авторизация успешна», а кнопки сбора данных и переключения режима отображения становятся активными. В случае ошибки выводится соответствующее сообщение с предложением проверить правильность учетных данных. Результат успешной авторизации представлен на рисунке 25.

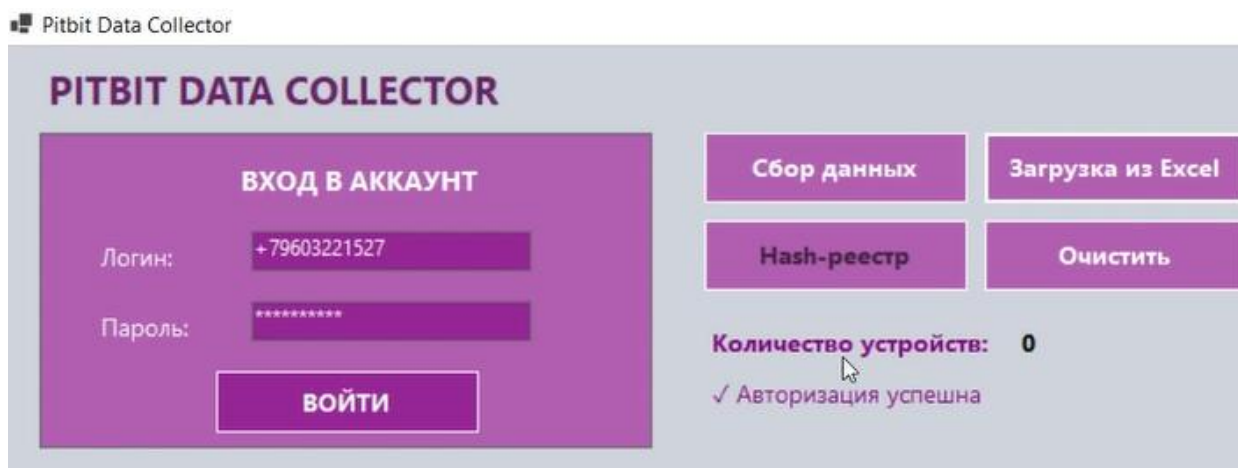


Рисунок 25 — Успешная авторизация

1.9.2 Сценарий «Сбор данных об устройствах»

После успешной авторизации пользователь может выполнить сбор актуальных данных о майнинговом оборудовании, нажав кнопку «Сбор данных» на панели управления. Процесс сбора данных начинается с перехода на главную страницу платформы Pitbit, после чего программа ожидает загрузки динамического контента. В строке состояния последовательно отображаются этапы выполнения: «Сбор данных...», «Загрузка данных о чипах...», а также информация о количестве обрабатываемых устройств.

Особенностью сбора данных является необходимость активации загрузки информации о состоянии чипов. Программа автоматически находит и нажимает все необходимые кнопки на странице, после чего выдерживает паузу для полной загрузки данных с сервера. После завершения парсинга HTML-кода страницы собранная информация отображается в центральной таблице программы. Ход выполнения операции визуализируется в строке

состояния и счетчике устройств. Результат сбора данных представлен на рисунке 26.

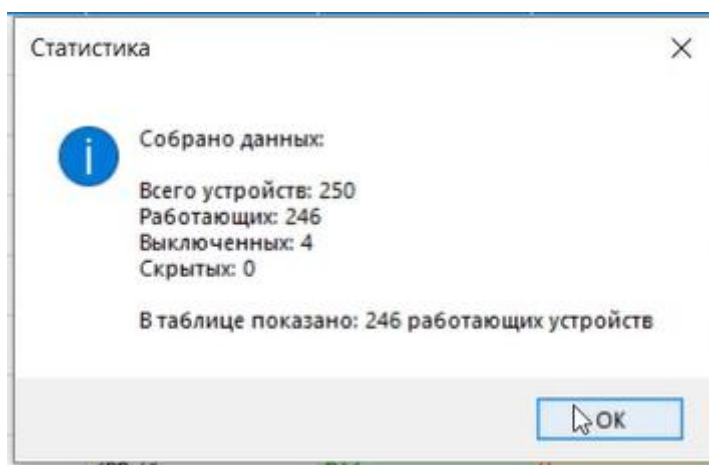


Рисунок 26 — Данные об устройствах после сбора

По завершении сбора данных пользователю отображается информационное окно со статистикой: общее количество найденных устройств, количество работающих, выключенных и скрытых устройств. В таблице по умолчанию отображаются только работающие устройства, что позволяет сосредоточиться на активном оборудовании.

1.9.3 Сценарий «Переключение режима отображения»

Для удобства анализа данных в программе реализована возможность переключения между режимами отображения устройств. По умолчанию активен режим отображения только работающих устройств, при этом кнопка переключения имеет надпись «Все устройства». При нажатии на данную кнопку в таблице отображаются все найденные устройства, включая выключенные и скрытые, а надпись на кнопке меняется на «Только работающие». Повторное нажатие возвращает режим отображения только активного оборудования. Счетчик устройств и строка состояния динамически обновляются, отображая текущее количество показываемых устройств. Пример переключения режимов представлен на рисунках 27,28.

PITBIT DATA COLLECTOR

ВХОД В АККАУНТ

Логин: +79603221527
Пароль: *****
ВОЙТИ

Сбор данных **Загрузка из Excel** **Все устройства** **Экспорт в Excel**

Hash-реестр **Очистить**

Количество устройств: 246
Показаны только работающие: 246

ID	Модель	IP Адрес	Воркер	Хэшрейт 5с (TH/s)	Хэшрейт 30мин	Чипы (хорошие)	Чипы (плохие)	Чипы (слом.)	Всего чипов	Мощность (W)	Температура (°C)	Статус
13470	S21 Hydov2	192.168.54.209	sergeymoskaev.workerLE...	350.52	354.6	648	0	0	648	5675	55	Работает
14743	S19 XP Hydov2	192.168.54.217	sergeymoskaev.workerLE...	277.68	271.2	612	0	0	612	5619	66	Работает
14745	S19 XP Hydov2	192.168.54.251	sergeymoskaev.workerLE...	270.04	268.36	612	0	0	612	5654	63	Работает
14974	S19 XP Hydov2	192.168.54.232	sergeymoskaev.workerLE...	278.67	273.75	612	0	0	612	5678	64	Работает
15151	S19 XP Hydov2	192.168.54.195	sergeymoskaev.workerLE...	268.22	270.85	612	0	0	612	5749	65	Работает
15332	S19 XP Hydov2	192.168.54.216	sergeymoskaev.workerLE...	241.23	242.08	608	2	2	612	5175	61	Работает
15399	S19 Pro Hydov2	192.168.54.100	sergeymoskaev.workerLE...	167.07	165.45	720	0	0	720	4029	59	Работает
15784	S19 XP Hydov2	192.168.54.234	sergeymoskaev.workerLE...	271.03	268.28	612	0	0	612	5680	65	Работает
15785	S19 XP Hydov2	192.168.54.133	sergeymoskaev.workerLE...	268.22	266.73	612	0	0	612	5671	63	Работает
18254	S21 Hydov2	192.168.54.157	sergeymoskaev.workerLE...	342.17	346.31	648	0	0	648	5506	55	Работает
18255	S21 Hydov2	192.168.54.152	sergeymoskaev.workerLE...	351.84	344.73	648	0	0	648	5336	56	Работает
18256	S21 Hydov2	192.168.54.156	sergeymoskaev.workerLE...	342.06	346.6	648	0	0	648	5467	55	Работает
18257	S21 Hydov2	192.168.54.179	sergeymoskaev.workerLE...	359.43	353.9	648	0	0	648	5509	56	Работает
18261	S21 Hydov2	192.168.55.22	sergeymoskaev.workerLE...	359.98	354.13	648	0	0	648	5590	55	Работает
18262	S21 Hydov2	192.168.55.26	sergeymoskaev.workerLE...	344.37	344.24	648	0	0	648	5562	55	Работает
18263	S21 Hydov2	192.168.55.23	sergeymoskaev.workerLE...	340.85	338.07	648	0	0	648	5502	54	Работает
18265	S21 Hydov2	192.168.55.27	sergeymoskaev.workerLE...	349.2	347.7	648	0	0	648	5478	55	Работает

Рисунок 27 — Отображение только работающих устройств

PITBIT DATA COLLECTOR

ВХОД В АККАУНТ

Логин: +79603221527
Пароль: *****
ВОЙТИ

Сбор данных **Загрузка из Excel** **Только работающие** **Экспорт в Excel**

Hash-реестр **Очистить**

Количество устройств: 250
Показаны все устройства: 250

ID	Модель	IP Адрес	Воркер	Хэшрейт 5с (TH/s)	Хэшрейт 30мин	Чипы	Чипы (плохие)	Чипы (слом.)	Всего чипов	Мощность (W)	Температура (°C)	Статус
13470	S21 Hydov2	192.168.54.209	sergeymoskaev.workerLE...	350.52	354.6	648	0	0	648	5675	55	Работает
14743	S19 XP Hydov2	192.168.54.217	sergeymoskaev.workerLE...	277.68	271.2	612	0	0	612	5619	66	Работает
14745	S19 XP Hydov2	192.168.54.251	sergeymoskaev.workerLE...	270.04	268.36	612	0	0	612	5654	63	Работает
14974	S19 XP Hydov2	192.168.54.232	sergeymoskaev.workerLE...	278.67	273.75	612	0	0	612	5678	64	Работает
15151	S19 XP Hydov2	192.168.54.195	sergeymoskaev.workerLE...	268.22	270.85	612	0	0	612	5749	65	Работает
15332	S19 XP Hydov2	192.168.54.216	sergeymoskaev.workerLE...	241.23	242.08	608	2	2	612	5175	61	Работает
15399	S19 Pro Hydov2	192.168.54.100	sergeymoskaev.workerLE...	167.07	165.45	720	0	0	720	4029	59	Работает
15784	S19 XP Hydov2	192.168.54.234	sergeymoskaev.workerLE...	271.03	268.28	612	0	0	612	5680	65	Работает
15785	S19 XP Hydov2	192.168.54.133	sergeymoskaev.workerLE...	268.22	266.73	612	0	0	612	5671	63	Работает
18254	S21 Hydov2	192.168.54.157	sergeymoskaev.workerLE...	342.17	346.31	648	0	0	648	5506	55	Работает
18255	S21 Hydov2	192.168.54.152	sergeymoskaev.workerLE...	351.84	344.73	648	0	0	648	5336	56	Работает
18256	S21 Hydov2	192.168.54.156	sergeymoskaev.workerLE...	342.06	346.6	648	0	0	648	5467	55	Работает
18257	S21 Hydov2	192.168.54.179	sergeymoskaev.workerLE...	359.43	353.9	648	0	0	648	5509	56	Работает
18261	S21 Hydov2	192.168.55.22	sergeymoskaev.workerLE...	359.98	354.13	648	0	0	648	5590	55	Работает
18262	S21 Hydov2	192.168.55.26	sergeymoskaev.workerLE...	344.37	344.24	648	0	0	648	5562	55	Работает
18263	S21 Hydov2	192.168.55.23	sergeymoskaev.workerLE...	340.85	338.07	648	0	0	648	5502	54	Работает
18265	S21 Hydov2	192.168.55.27	sergeymoskaev.workerLE...	349.2	347.7	648	0	0	648	5478	55	Работает

Рисунок 28 — Отображение всех устройств

1.9.4 Сценарий «Просмотр детальной информации об устройстве»

Для получения подробной информации о конкретном устройстве пользователь может дважды щелкнуть левой кнопкой мыши по соответствующей строке в таблице. При двойном щелчке программа

формирует информационное окно, содержащее все доступные параметры выбранного майнера: идентификатор, модель, IP-адрес, имя воркера, значения хэшрейта за 5 секунд и 30 минут, количество чипов различного состояния, общее количество чипов, потребляемую мощность, температуру и текущий статус. Данная функция позволяет оперативно оценить состояние отдельного устройства без необходимости открытия веб-интерфейса. Пример отображения детальной информации представлен на рисунке 29.

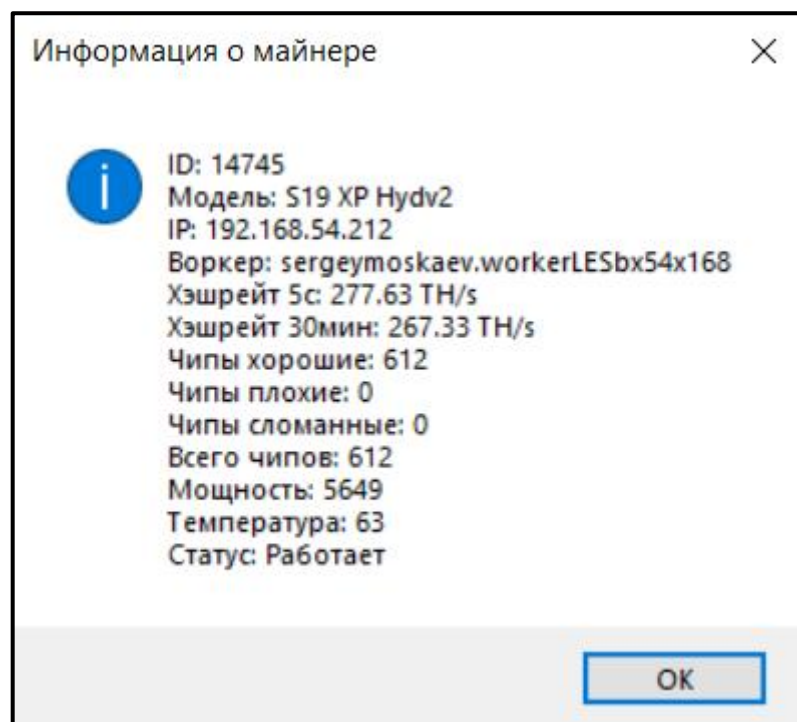


Рисунок 29 — Окно детальной информации об устройстве

1.9.5 Сценарий «Экспорт данных в Excel»

Для сохранения собранных данных и последующего анализа в программе предусмотрена функция экспорта в формат Excel, вызываемая нажатием кнопки «Экспорт в Excel». При нажатии кнопки открывается диалоговое окно сохранения файла, в котором автоматически формируется имя файла, включающее тип отображаемых устройств (работающие или все) и текущую дату со временем, как показано на рисунке 30.

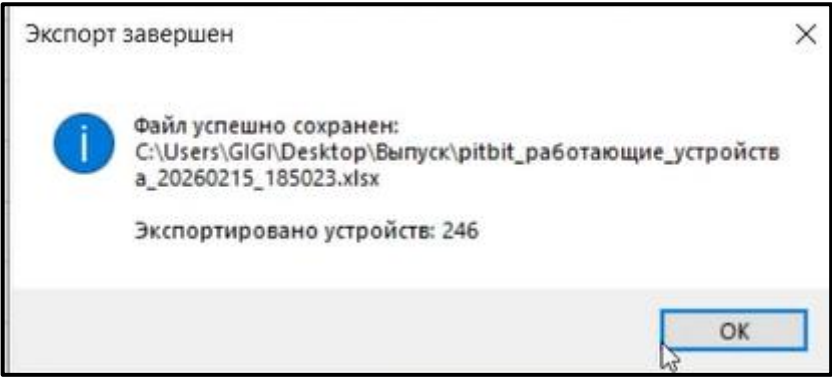


Рисунок 30 — Диалог сохранения Excel-файла

После выбора пути сохранения программа создает файл Excel с форматированной таблицей. Заголовки столбцов выделяются жирным начертанием и серым фоном. Строки с данными окрашиваются в зависимости от статуса устройства: зеленый цвет для работающих устройств, серый для выключенных, светло-красный для скрытых. Ширина столбцов автоматически подбирается под содержимое. После завершения экспорта отображается информационное окно с указанием количества экспортированных устройств и пути к сохраненному файлу. Результат экспорта представлен на рисунке 31.

ID	Модель	IP Адрес	Воркер	Хэшрейт 5с (TH/s)	Хэшрейт 30мин (TH/s)	Чипы хорошие	Чипы плохие	Чипы сломанные	Всего чипов	Мощность (W)	Температура (°C)	Статус
11940	S19 Pro Hyd2	192.168.54.104	sergeymoskaev.workerLESbx54x179	0	0	0	0	0	0	Н/Д	40	Работает
13470	S21 Hyd2	192.168.54.152	sergeymoskaev.workerLESbx54x138	349.75	356.06	648	0	0	648	5714	55	Работает
14473	S19 Pro Hyd2	192.168.54.15	sergeymoskaev.workerES00x54x15	141.18	144.55	540	0	0	540	4080	69	Работает
14743	S19 XP Hyd2	192.168.54.162	sergeymoskaev.workerLESbx54x148	271.03	270.21	612	0	0	612	5599	66	Работает
14745	S19 XP Hyd2	192.168.54.212	sergeymoskaev.workerLESbx54x168	277.63	267.33	612	0	0	612	5649	63	Работает
14873	S19 Pro Hyd2	192.168.54.189	sergeymoskaev.workerLESbx54x160	210	206.83	720	0	0	720	5841	61	Работает
14875	S19 Pro Hyd2	192.168.54.224	sergeymoskaev.workerLESbx54x181	0	0	0	0	0	0	Н/Д	39	Работает
14883	S19 Pro Hyd2	192.168.54.101	sergeymoskaev.workerLESbx54x43	0	0	0	0	0	0	Н/Д	40	Работает
14887	S19 Pro Hyd2	192.168.54.84	sergeymoskaev.workerLESbx54x32	211.38	209.34	720	0	0	720	5838	64	Работает
14900	S19 Pro Hyd2	192.168.54.188	sergeymoskaev.workerLESbx54x159	213.3	207.4	720	0	0	720	5879	63	Работает
14903	S19 Pro Hyd2	192.168.54.110	sergeymoskaev.workerLESbx54x20	0	0	0	0	0	0	Н/Д	40	Работает
14908	S19 Pro Hyd2	192.168.54.196	sergeymoskaev.workerLESbx54x167	0	0	0	0	0	0	Н/Д	40	Работает
14922	S19 Pro Hyd2	192.168.54.99	sergeymoskaev.workerLESbx54x40	0	0	0	0	0	0	Н/Д	40	Работает
14927	S19 Pro Hyd2	192.168.54.91	sergeymoskaev.workerLESbx54x18	0	0	0	0	0	0	Н/Д	40	Работает
14938	S19 Pro Hyd2	192.168.54.90	sergeymoskaev.workerLESbx54x25	210.06	210.82	720	0	0	720	5856	67	Работает
14940	S19 Pro Hyd2	192.168.54.83	sergeymoskaev.workerLESbx54x31	0	0	0	0	0	0	Н/Д	40	Работает
14949	S19 Pro Hyd2	192.168.54.159	sergeymoskaev.workerLESbx54x58	0	0	0	0	0	0	Н/Д	40	Работает
14974	S19 XP Hyd2	192.168.54.192	sergeymoskaev.workerLESbx54x163	275.48	273.32	612	0	0	612	5654	64	Работает
15151	S19 XP Hyd2	192.168.54.131	sergeymoskaev.workerLESbx54x132	270.37	267.15	608	4	0	612	5670	65	Работает
15245	S19 Pro Hyd2	192.168.54.112	sergeymoskaev.workerLESbx54x42	0	0	0	0	0	0	Н/Д	40	Работает
15252	S19 Pro Hyd2	192.168.54.98	sergeymoskaev.workerLESbx54x40	0	0	0	0	0	0	Н/Д	40	Работает
15271	S19 Pro Hyd2	192.168.54.79	sergeymoskaev.workerLESbx54x41	218.42	213.54	720	0	0	720	5897	65	Работает
15292	S19 Pro Hyd2	192.168.54.117	sergeymoskaev.workerLESbx54x56	215.06	211.55	720	0	0	720	5835	64	Работает
15299	S19 Pro Hyd2	192.168.54.78	sergeymoskaev.workerLESbx54x64	222.27	218.05	720	0	0	720	5938	63	Работает
15307	S19 Pro Hyd2	192.168.54.193	sergeymoskaev.workerLESbx54x164	210.94	211.53	720	0	0	720	5851	65	Работает
15316	S19 Pro Hyd2	192.168.54.114	sergeymoskaev.workerLESbx54x55	204.78	205.92	720	0	0	720	5804	67	Работает
15321	S19 Pro Hyd2	192.168.54.97	sergeymoskaev.workerLESbx54x44	185.21	189.07	540	0	0	540	5864	70	Работает
15334	S19 Pro Hyd2	192.168.54.118	sergeymoskaev.workerLESbx54x53	210.94	209.48	720	0	0	720	5824	65	Работает

Рисунок 31 — Сохраненный Excel-файл

1.9.6 Сценарий «Экспорт в Hash-реестр»

Для подготовки данных в специализированном формате, используемом на платформе Pitbit, в программе реализована функция экспорта в hash-реестр, вызываемая нажатием кнопки «Hash-реестр». При нажатии кнопки открывается диалог выбора файла шаблона Excel, как показано на рисунке 32.

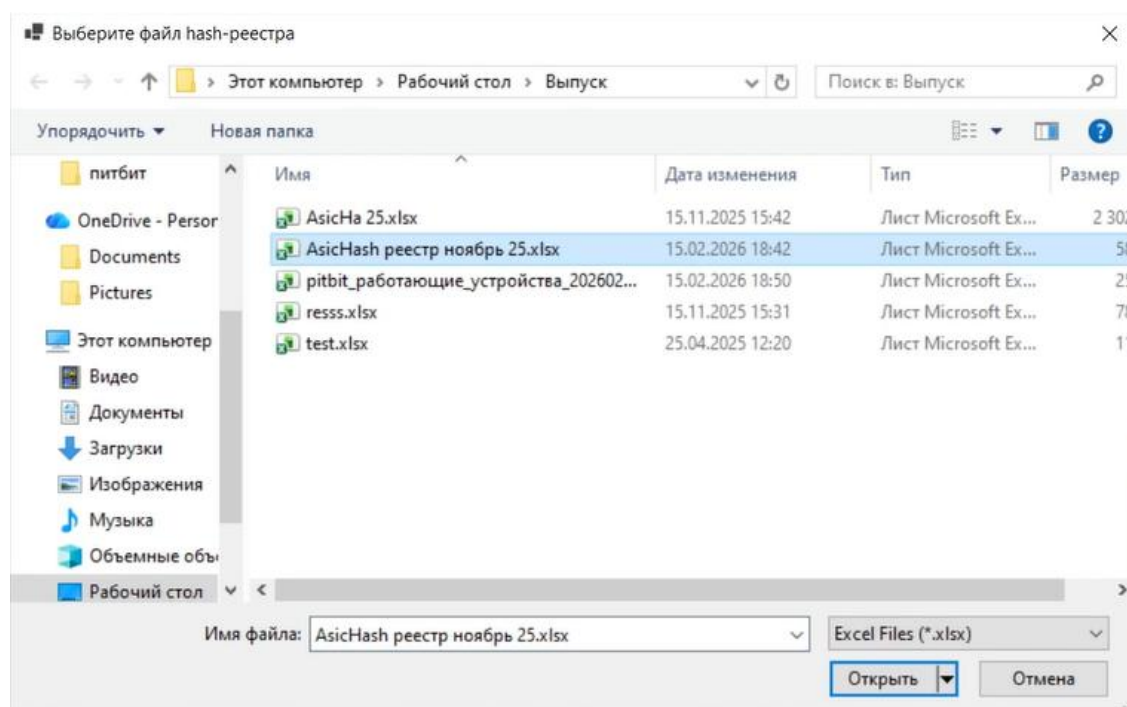


Рисунок 32 — Выбор файла шаблона hash-реестра

После выбора файла пользователю предлагается выбрать лист из числа доступных в книге, как показано на рисунке 33.

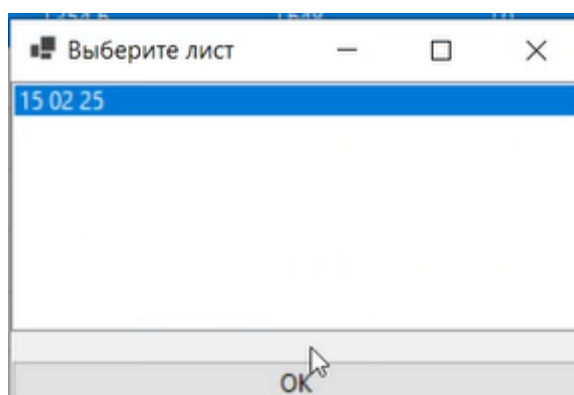


Рисунок 33 — Выбор листа в файле шаблона

Программа автоматически распределяет устройства по заданным диапазонам ячеек в зависимости от имени воркера. Устройства с префиксами workerlesbx и workerlesax размещаются в столбцах В-F, устройства с префиксом workerles00x — в столбцах Н-L. Внутри каждого диапазона устройства сортируются по убыванию значения хэшрейта. После заполнения данных обновляется статистическая информация о количестве устройств различных моделей с разбивкой по количеству исправных чипов. По завершении экспорта отображается сводная информация о количестве вставленных устройств и статистике по моделям. Результат заполнения шаблона представлен на рисунке 34.

Количество словачных ячеек			S19	S19Exp	S21	S19 Pro	2 платы	2 платы	1 плата	S19Exp	2 платы	1 плата	S21	2 платы	1 плата
241 000000			241	1 465 000	0004 101877	00 316 7	1 465 000	22,7	1,0	0000 0000	241	227	0	0	0
Стойка			199	200 000	61 622,3	44	174,28	7 689,9	коп-во чипов						
192.168.54.136	S21 Hydr02	350970	351460	648	192.168.54.106	S19 XP Hydr02	240560	252360	612						
192.168.54.209	S21 Hydr02	350530	351600	648	192.168.54.118	S19 XP Hydr02	205000	218010	197						
192.168.54.80	S21 Hydr02	346790	354510	648	192.168.54.105	S19 XP Hydr02	207260	207060	612						
192.168.55.22	S21 Hydr02	359880	354130	648	192.168.54.116	S19 XP Hydr02	206320	206620	612						
192.168.54.69	S21 Hydr02	357010	354090	648	192.168.54.119	S19 XP Hydr02	205440	206000	612						
192.168.54.179	S21 Hydr02	359430	353900	648	192.168.54.104	S19 XP Hydr02	204670	206330	612						
192.168.54.134	S21 Hydr02	355470	353320	648	192.168.54.102	S19 XP Hydr02	206380	206110	612						
192.168.54.51	S21 Hydr02	350850	353310	648	192.168.54.210	S19 XP Hydr02	198680	205870	612						
192.168.54.62	S21 Hydr02	359430	353900	648	192.168.54.117	S19 XP Hydr02	200220	205380	612						
192.168.54.187	S21 Hydr02	345680	352470	648	192.168.54.101	S19 XP Hydr02	202860	204170	612						
192.168.54.153	S21 Hydr02	365040	352210	648	192.168.54.99	S19 XP Hydr02	201870	203650	612						
192.168.54.66	S21 Hydr02	354370	351700	648	192.168.54.124	S19 XP Hydr02	205720	202210	612						
192.168.54.176	S21 Hydr02	350630	351600	648	192.168.54.86	S19 XP Hydr02	200680	202080	612						
192.168.54.55	S21 Hydr02	354480	351170	648	192.168.54.121	S19 XP Hydr02	200490	201860	612						
192.168.54.180	S21 Hydr02	355360	351160	648	192.168.54.122	S19 XP Hydr02	204730	199150	612						
192.168.54.41	S21 Hydr02	347990	351160	648	192.168.54.114	S19 XP Hydr02	201480	198360	612						
192.168.54.158	S21 Hydr02	368010	350790	648	192.168.54.109	S19 XP Hydr02	195710	198220	612						
192.168.55.19	S21 Hydr02	351510	350580	648	192.168.54.115	S19 XP Hydr02	202970	198200	612						
192.168.54.166	S21 Hydr02	351180	350250	648	192.168.54.98	S19 XP Hydr02	190980	197280	612						
192.168.54.37	S21 Hydr02	347220	350290	648	192.168.54.123	S19 XP Hydr02	193350	196390	612						
192.168.54.39	S21 Hydr02	353490	350140	647	192.168.54.112	S19 XP Hydr02	191970	195550	612						
192.168.54.186	S21 Hydr02	348100	350040	648	192.168.54.97	S19 XP Hydr02	197970	195340	612						
192.168.54.183	S21 Hydr02	337990	349970	648	192.168.54.111	S19 XP Hydr02	193240	196030	612						
192.168.54.159	S21 Hydr02	343490	349960	648	192.168.54.120	S19 XP Hydr02	183510	186920	408						
192.168.54.38	S21 Hydr02	350300	348800	648	192.168.54.12	S19 Pro Hydr02	168110	168990	720						
192.168.54.72	S21 Hydr02	353050	349650	648	192.168.54.103	S19 Pro Hydr02	171360	167330	720						
192.168.54.46	S21 Hydr02	332270	349540	648	192.168.54.13	S19 Pro Hydr02	166300	167500	720						
192.168.54.170	S21 Hydr02	357670	348480	648	192.168.54.10	S19 Pro Hydr02	166190	167110	720						
192.168.54.45	S21 Hydr02	349200	348470	648	192.168.54.26	S19 Pro Hydr02	169760	166800	720						
192.168.54.173	S21 Hydr02	345910	348460	648	192.168.54.11	S19 Pro Hydr02	167120	165990	720						
192.168.55.12	S21 Hydr02	346790	348110	648	192.168.54.100	S19 Pro Hydr02	167070	165450	720						
192.168.54.182	S21 Hydr02	356790	349050	648	192.168.54.113	S19 Pro Hydr02	163330	164860	720						
192.168.54.248	S21 Hydr02	349750	349020	648	192.168.54.14	S19 Pro Hydr02	165200	164830	720						
192.168.54.149	S21 Hydr02	354810	349010	648	192.168.54.27	S19 Pro Hydr02	160030	164120	720						
192.168.54.58	S21 Hydr02	349310	349000	648	192.168.54.107	S19 Pro Hydr02	161680	163800	720						
192.168.54.154	S21 Hydr02	346560	348500	648	192.168.54.34	S19 Pro Hydr02	161240	163110	720						
192.168.54.36	S21 Hydr02	343930	348470	648	192.168.54.110	S19 Pro Hydr02	162950	162500	720						
192.168.54.49	S21 Hydr02	344370	348410	648	192.168.54.29	S19 Pro Hydr02	165920	162410	720						
192.168.54.198	S21 Hydr02	351070	348330	648	192.168.54.108	S19 Pro Hydr02	156030	161930	720						
192.168.54.21	S21 Hydr02	362620	348120	648	192.168.54.28	S19 Pro Hydr02	159430	161600	720						
192.168.54.181	S21 Hydr02	344590	347870	648	192.168.54.31	S19 Pro Hydr02	159920	161230	720						
192.168.54.53	S21 Hydr02	340190	347840	648	192.168.54.33	S19 Pro Hydr02	161350	158880	720						
192.168.54.167	S21 Hydr02	356130	347740	648	192.168.54.24	S19 Pro Hydr02	157060	157160	720						
192.168.55.27	S21 Hydr02	349200	347700	648	192.168.54.32	S19 Pro Hydr02	158550	157040	720						
192.168.55.20	S21 Hydr02	337440	347600	648							7 689,9		7 689,9		
192.168.55.24	S21 Hydr02	344820	347480	648							200 000		200 001		
192.168.54.25	S21 Hydr02	353930	347460	648							20,08 ESP		20,08 ESP		
192.168.54.240	S21 Hydr02	340020	347380	648											
192.168.54.174	S21 Hydr02	344590	347100	648											
192.168.54.240	S21 Hydr02	339200	347040	648											
192.168.54.52	S21 Hydr02	355470	347020	648											

Рисунок 34 — Заполненный Hash-реестр

1.9.7 Сценарий «Загрузка данных из Excel»

В программе предусмотрена возможность загрузки ранее сохраненных данных из Excel-файлов, что позволяет работать с архивными снимками состояния оборудования без повторного обращения к веб-интерфейсу. Функция вызывается нажатием кнопки «Загрузка из Excel». При нажатии открывается диалог выбора файла, как показано на рисунке 35.

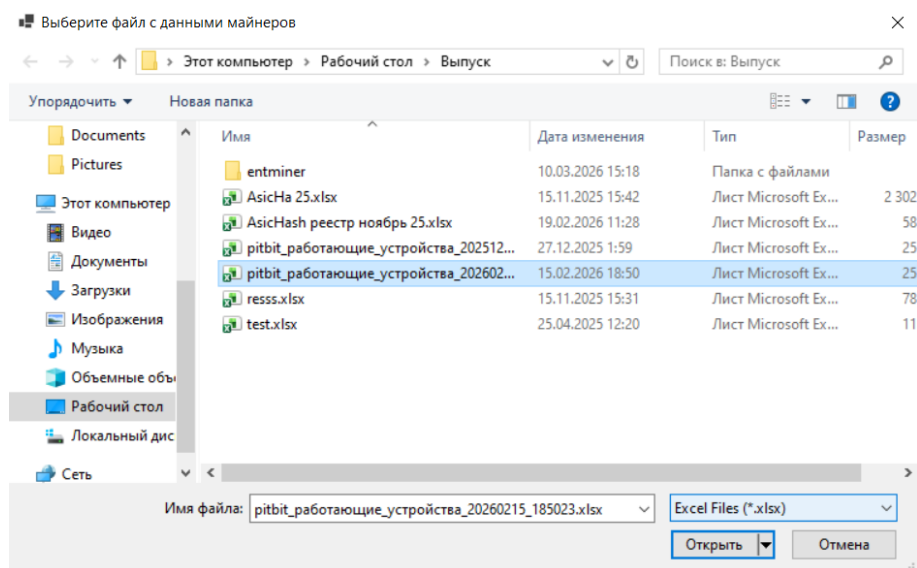


Рисунок 35 — Выбор файла для загрузки данных

После выбора файла программа считывает данные с первого листа, начиная со второй строки, и заполняет внутренние списки данных. Загруженная информация отображается в таблице с сохранением цветового форматирования в зависимости от статуса устройств. По завершении загрузки отображается информационное окно со статистикой: общее количество загруженных устройств, количество работающих, выключенных и скрытых. Результат загрузки представлен на рисунке 36.

PITBIT DATA COLLECTOR

ВХОД В АККАУНТ

Логин: +79603221527

Пароль: [скрыт]

ВОЙТИ

Сбор данных **Загрузка из Excel** **Все устройства** **Экспорт в Excel**

Hash-реестр **Очистить**

Количество устройств: 246

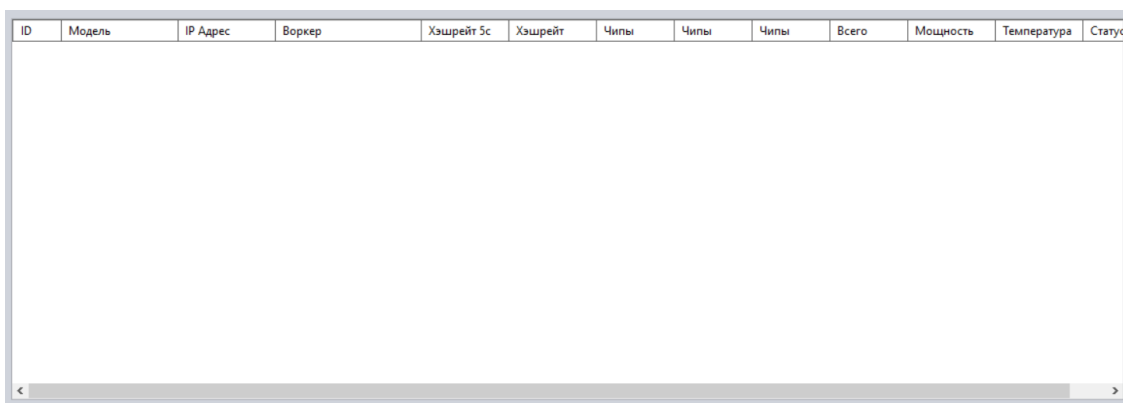
✓ Загружено из Excel: 246 устройств

ID	Модель	IP Адрес	Воркер	Хэшрейт 5с	Хэшрейт	Чипы	Чипы	Чипы	Всего	Мощность	Температура	Статус
13470	S21 Hydov2	192.168.54.209	sergeymoskaev.workerLE...	350.52	354.6	648	0	0	648	5675	55	Раб
14743	S19 XP Hydov2	192.168.54.217	sergeymoskaev.workerLE...	277.68	271.2	612	0	0	612	5619	66	Раб
14745	S19 XP Hydov2	192.168.54.251	sergeymoskaev.workerLE...	270.04	268.36	612	0	0	612	5654	63	Раб
14974	S19 XP Hydov2	192.168.54.232	sergeymoskaev.workerLE...	278.67	273.75	612	0	0	612	5678	64	Раб
15151	S19 XP Hydov2	192.168.54.195	sergeymoskaev.workerLE...	268.22	270.85	612	0	0	612	5749	65	Раб
15332	S19 XP Hydov2	192.168.54.216	sergeymoskaev.workerLE...	241.23	242.08	608	2	2	612	5175	61	Раб
15399	S19 Pro Hydov2	192.168.54.100	sergeymoskaev.workerLE...	167.07	165.45	720	0	0	720	4029	59	Раб
15784	S19 XP Hydov2	192.168.54.234	sergeymoskaev.workerLE...	271.03	268.28	612	0	0	612	5680	65	Раб
15785	S19 XP Hydov2	192.168.54.133	sergeymoskaev.workerLE...	268.22	266.73	612	0	0	612	5671	63	Раб
18254	S21 Hydov2	192.168.54.157	sergeymoskaev.workerLE...	342.17	346.31	648	0	0	648	5506	55	Раб
18255	S21 Hydov2	192.168.54.152	sergeymoskaev.workerLE...	351.84	344.73	648	0	0	648	5336	56	Раб
18256	S21 Hydov2	192.168.54.156	sergeymoskaev.workerLE...	342.06	346.6	648	0	0	648	5467	55	Раб

Рисунок 36 — Данные после загрузки из Excel

1.9.8 Сценарий «Очистка данных»

Для сброса текущих данных и подготовки программы к новому сбору информации предусмотрена функция очистки, вызываемая нажатием кнопки «Очистить». При нажатии кнопки очищается таблица, сбрасываются внутренние списки данных, строка состояния возвращается к исходному значению «Готов к работе», счетчик устройств обнуляется. Кнопки экспорта и переключения режима отображения деактивируются до момента загрузки новых данных. Результат очистки представлен на рисунке 37.



ID	Модель	IP Адрес	Воркер	Хэшрейт 5с	Хэшрейт	Чипы	Чипы	Чипы	Всего	Мощность	Температура	Статус

Рисунок 37 — Программа после очистки данных

1.9.9 Дополнительные возможности

Также описаны и дополнительные функции приложения:

- адаптивность окна. Программа поддерживает изменение размеров главного окна. При изменении размеров автоматически пересчитываются положения и размеры панелей, кнопок и таблицы, что обеспечивает корректное отображение интерфейса при любых размерах окна;
- цветовая индикация в таблице. Для улучшения визуального восприятия в таблице применяется цветовое оформление. Работающие устройства выделяются зеленым фоном, выключенные — серым, скрытые — светло-красным. Ячейки с количеством чипов также имеют цветовую индикацию: исправные чипы на зеленом фоне, предупредительные — на

желтом, сломанные — на красном. Пример цветового оформления представлен на рисунке 38;

612	0	0
612	0	0
608	2	2
720	0	0

Рисунок 38 — Цветовая индикация в таблице

– строка состояния. В нижней части панели управления расположена строка состояния, отображающая текущий статус выполнения операций, результаты завершенных действий и сообщения об ошибках. Справа от строки состояния отображается счетчик устройств, показывающий количество элементов в текущем представлении таблицы;

– счетчик устройств. Над строкой состояния расположена метка «Количество устройств:» и соответствующее числовое значение, которое динамически обновляется при изменении данных в таблице или переключении режима отображения. На рисунке 39 видно расположение счетчика.

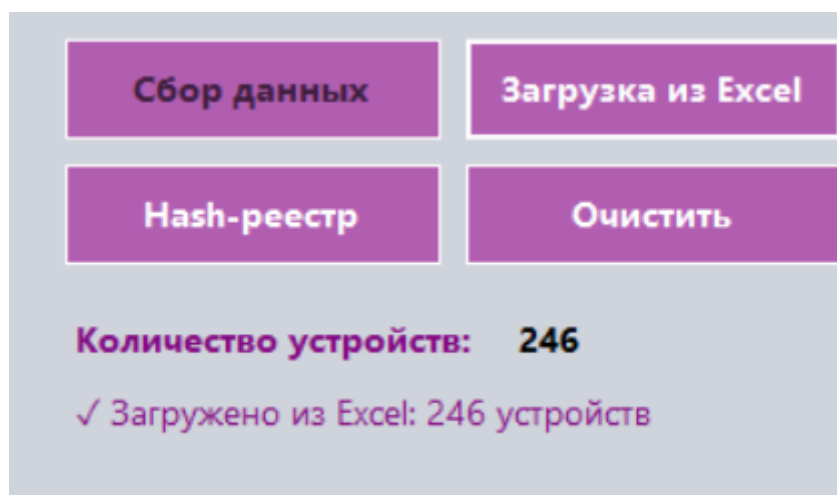


Рисунок 39 — Цветовая индикация в таблице

2 Требования безопасности по охране труда при работе с ПЭВМ.

В данном разделе расписаны требования по обеспечению безопасности жизнедеятельности и охране труда при работе с персональными компьютерами, при разработке ПО, а также методы снижения вредного воздействия на здоровье человека.

Трудовой процесс осуществляется в определенных условиях производственной среды, которые характеризуются совокупностью её элементов и факторов, которые влияют на трудоспособность и состояние здоровья человека в процессе работы. Производственная среда и факторы трудового процесса составляют в совокупности условия работы.

Площадь на одно рабочее место с ПК (монитор с электронно-лучевой трубкой) должна составлять не менее 6 м², с монитором на базе плоских экранов (жидкокристаллические, плазменные) - 4,5м², а объем - не менее 20,0м³ [4]. Не следует размещать рабочие места с ПК вблизи силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе ПК. Полимерные материалы, используемые для внутренней отделки интерьера помещений, где имеются ПК, не должны изменять своих химических свойств в течение всего срока эксплуатации, не позволять накапливаться статическому электричеству. Запрещено использование полимерных материалов (древесностружечные плиты, слоистый бумажный пластик, синтетические ковровые покрытия и др.), выделяющих в воздух вредные химические вещества. В помещениях ежедневно должна проводиться влажная уборка [4].

Помещения для эксплуатации ПК должны иметь естественное (боковое или верхнее освещение помещений естественным светом) и искусственное (совмещенное - освещение, при котором недостаточное по нормам

естественное освещение дополняется искусственным, за счет дополнительных световых приборов) освещение [4].

2.1 Организация рабочего места

Основные причины ухудшения самочувствия при работе за компьютером носят неспецифический характер: гиподинамия с нарушением кровообращения, нефизиологичное положение тела, повторяющиеся однообразные движения, а также воздействие электромагнитного излучения, статического электричества и неблагоприятного микроклимата. Наиболее серьёзные заболевания связаны с костно-мышечной системой, зрением и обострениями сопутствующих патологий. Работа протекает в условиях высокой нервно-психической напряжённости при длительном нахождении в вынужденной позе [5].

Шейный отдел позвоночника наиболее подвижен — при его анатомо-физиологических изменениях нарушается кровоснабжение головного мозга (головные боли, головокружение, «мушки» перед глазами, шаткость походки) [5]. В условиях, когда эргономика рабочего места не соответствует физиологическим нормам, биомеханический фактор становится значимым наравне с прочими вредными факторами, однако его действие может быть уменьшено самим работающим за счёт использования правильной позы и движений [5].

Требования к компьютерной мебели:

- столешница должна обеспечивать достаточное место для размещения клавиатуры, мыши, документов;
- высота стола — не менее 74 см;
- возможность размещения компонентов ПК на разных уровнях;
- наличие дополнительной поверхности для клавиатуры и мыши, расположенной ниже основной;
- системный блок размещается в специальном отсеке под столом.

Наиболее эффективную зону охвата обеспечивает полукруглый стол, когда пользователь находится в центре полуокружности. Обычный офисный стол создаёт вынужденное положение шеи и большое расстояние до монитора. Г-образный стол оставляет значительную часть поверхности вне зоны доступа.

Клавиатуру необходимо располагать в 10–15 см от края стола, чтобы локти находились параллельно поверхности стола, под прямым углом к плечу, а локоть и кисть составляли одну линию. Это переносит нагрузку на предплечье, предотвращая повреждения лучезапястного сустава.

Монитор должен находиться на расстоянии не менее вытянутой руки. При полукруглом столе монитор оптимально размещается в зоне охвата. Важна частота вертикальной развёртки — менее 75 Гц приводит к быстрому утомлению глаз.

Работа с компьютером связана с напряжением зрения. Мышцы, фокусирующие глаз, устают от чрезмерной нагрузки. При длительной работе за монитором глаза «замирают», мышцы находятся в постоянном напряжении, что ведёт к изменению формы хрусталика, головокружению и снижению внимания.

Лучезапястный сустав подвергается риску при неправильном использовании клавиатуры и мыши. Запястный канал узок — при отёке сдавливается срединный нерв, что вызывает расстройства чувствительности и движения. При работе с мышью кисть не должна быть изогнута в запястье; перемещать мышь рекомендуется движениями пальцев, удерживая запястье на столе. Мышь подбирается индивидуально под размер руки пользователя.

2.2 Освещение рабочего места

Помещения должны иметь естественное и искусственное освещение. Согласно действующим санитарным правилам СП 2.2.3670-20, показатели освещённости не должны превышать установленных гигиенических нормативов для соответствующих видов работ. Ранее действовавшие нормы

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		54

(СанПиН 2.2.2/2.4.1340-03) рекомендовали освещенность на поверхности стола 300–500 лк и ограничивали освещенность экрана до 300 лк. Контроль за освещенностью обязателен, если по результатам спецоценки условия труда признаны вредными по данному фактору.

2.3 Электробезопасность

Помещения, предназначенные для эксплуатации персональных компьютеров, согласно Правилам устройства электроустановок, как правило, классифицируются как помещения без повышенной опасности. Данная классификация справедлива при условии отсутствия токопроводящей пыли, высокой влажности воздуха, химически активных сред и токопроводящих полов. Тем не менее, высокая плотность размещения электрооборудования формирует комплекс факторов риска, пренебрежение которыми недопустимо [6].

К числу физически опасных и вредных производственных факторов при работе с ПЭВМ относятся: наличие в электрической цепи опасного для жизни напряжения (220 В), электромагнитные поля промышленной частоты (50 Гц), а также статическое электричество, аккумулируемое на диэлектрических поверхностях корпусов устройств [6].

Конструктивное исполнение оборудования должно исключать возможность случайного прикосновения пользователя к токоведущим частям. Все доступные металлические нетоковедущие части, которые могут оказаться под напряжением вследствие пробоя изоляции, должны иметь надёжное защитное заземление или зануление. Эксплуатация оборудования с механическими дефектами корпусов, изоляции кабелей, а также штепсельных соединений со следами оплавления или нагара категорически запрещена.

В организационном аспекте к самостоятельной работе с вычислительной техникой допускаются лица, прошедшие инструктаж по электробезопасности на I группу допуска [6]. Правилами категорически воспрещается: оставлять

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		55

оборудование под напряжением без присмотра; производить самостоятельное вскрытие кожухов блоков питания и мониторов; размещать электрические провода в зонах движения персонала без защитных коробов либо под напольными покрытиями, создавая механическую нагрузку на изоляцию. При визуальном обнаружении неисправности (искрение, дым, запах гари, ощущение протекания тока при касании) оборудование должно быть немедленно обесточено путём отключения от сети.

Отдельного внимания требует защита от статического электричества. Оптимальным методом профилактики накопления электростатических зарядов является поддержание в рабочем помещении относительной влажности воздуха на уровне 40–60%, применение антистатических покрытий, а также систематическое проведение влажной уборки.

2.4 Пожарная безопасность

Эксплуатация ПЭВМ сопряжена с горючей нагрузкой, формируемой изоляционными материалами кабельной продукции, полимерными деталями корпусов устройств и бумажными носителями информации. Потенциальными источниками зажигания выступают искрящие контакты, перегретые компоненты электронных схем и блоков питания. В соответствии с Федеральным законом № 123-ФЗ, оснащение рабочих помещений средствами противопожарной автоматики и оповещения закладывается на этапе проектирования или реконструкции объекта и находится в зоне ответственности эксплуатирующей организации.

Помещения, оборудованные компьютерной техникой, должны быть укомплектованы первичными средствами пожаротушения. Преимущество отдаётся углекислотным и порошковым огнетушителям, которые допускают тушение электроустановок без снятия напряжения. Планы эвакуации и световые указатели путей выхода должны размещаться на видных местах, а

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		56

сами пути эвакуации обязаны быть постоянно свободны для беспрепятственного прохода.

Пользователь со своей стороны принимает на себя обязательства по соблюдению норм противопожарного режима. В перечень запретов входит: использование в помещении открытого огня и курение вне специально отведённых мест; применение неисправных или несертифицированных удлинителей и тройников; расположение легковоспламеняющихся материалов и бумажных завалов на корпусах системных блоков и в непосредственной близости от сетевых фильтров; оставление работающей техники, включённой по завершении трудового дня без производственной необходимости.

В случае обнаружения очага возгорания работнику предписывается: незамедлительно обесточить аварийный участок сети; оповестить о происшествии голосом окружающих и вызвать подразделение пожарной охраны (101); при отсутствии угрозы жизни предпринять попытку локализации пламени штатным огнетушителем; при невозможности быстрой ликвидации горения — покинуть опасную зону, плотно закрыв за собой дверь, и действовать в строгом соответствии с планом эвакуации.

2.5 Режим труда и отдыха

Трудовая деятельность с использованием ПЭВМ характеризуется выраженным психофизиологическим напряжением. В структуре неблагоприятных функциональных изменений ведущую роль играют зрительное утомление и статическая нагрузка на опорно-двигательный аппарат, обусловленные длительным пребыванием в вынужденной рабочей позе и локальной гиподинамией. Несмотря на утрату силы Типовой инструкции ТОИ Р-45-084-01, регламентировавшей ранее нормативы отдыха для операторов ПЭВМ, сохранение профилактических мероприятий, базирующихся на физиологических принципах, остаётся актуальным и

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		57

закрепляется локальными нормативными актами работодателя либо личной самодисциплиной пользователя [7].

В целях нормирования трудовой нагрузки общепринято выделять три категории работ:

– группа А: деятельность по преимущественному считыванию информации с экрана, сопровождающаяся длительной фиксацией аккомодационного аппарата глаз;

– группа Б: интенсивный ввод информации с клавиатуры и координатных устройств, сопряжённый с высокой частотой стереотипных движений кистей;

– группа В: творческая работа в интерактивном режиме диалога с машиной, отличающаяся комбинированным нервно-эмоциональным и сенсорным напряжением [7].

В соответствии с физиолого-гигиеническими рекомендациями продолжительность непрерывной работы с дисплеем без регламентированного перерыва не должна превышать двух часов. Оптимальной схемой профилактики кумулятивного утомления является введение коротких пауз длительностью 10–15 минут после каждого 50-минутного или часового рабочего цикла. Суммарное время, отводимое на регламентированный отдых при стандартной восьмичасовой продолжительности смены, должно варьироваться в диапазоне от 30 до 70 минут в зависимости от категории и напряжённости труда [7].

Содержательное наполнение перерывов имеет принципиальное значение. В структуру пауз целесообразно включать следующие мероприятия:

– смена визуальной обстановки с выходом из зоны действия электромагнитных полей;

– выполнение комплекса упражнений для глаз: перевод фокуса с ближней точки на дальнюю, круговые движения глазных яблок при сомкнутых веках, лёгкий пальцевой массаж глазниц;

- проведение динамических упражнений для разгрузки шейного отдела позвоночника, плечевого пояса и мышц-сгибателей кисти;
- сквозное проветривание рабочей зоны для нормализации аэроионного состава воздуха.

В долгосрочной перспективе профилактика компьютерного синдрома требует от пользователя осознанного подхода к чередованию видов деятельности. Рекомендуется избегать монотонных операций в течение полного рабочего дня, при возможности комбинируя работу за терминалом с задачами, не требующими зрительной фиксации на экране.

Заключение

Разработка программы для составления отчётности майнинг-структуры представляет собой уникальное решение в области добычи криптовалюты. В условиях современного общества, где доступ к информации и увеличение энергоэффективности становятся критически важными, такой бот может стать надёжным помощником для майнинг-ферм, стремящихся улучшить производительность и увеличить прибыль. Интуитивный интерфейс и функциональные возможности программы создадут более благоприятные условия как для одиночных майнеров, так и для крупных промышленных ферм. Ожидается, что система не только упростит работу с большими данными, но и покажет на нюансы, над которыми стоит поработать. В дальнейшем проект может быть расширен путем добавления дополнительных функций и их логики взаимодействия.

					<i>ПГУ1.090301.18.001 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		60

Список использованных источников

1. Bixbit.tech [Электронный ресурс] URL: <https://bixbit.tech/ru> (дата обращения 17.05.2026)

2. Современный инновационный язык программирования с открытым исходным кодом для создания любых ваших приложений. [Электронный ресурс] Режим доступа: <https://dotnet.microsoft.com/ru-ru/languages/csharp> свободный. (дата обращения: 25.05.2026).

3. Мощная интегрированная среда разработки Visual Studio [Электронный ресурс] / Режим доступа: <https://visualstudio.microsoft.com/ru/vs/> свободный (дата обращения: 01.05.2026).

4. Ефремова О. С. Требования охраны труда при работе на персональных электронно-вычислительных машинах. – 2-е изд., перераб. и доп // М.: Издательство «Альфа-Пресс. – 2008.: [Электронный ресурс]. URL: http://www.bestbook.ru/userfiles/books/pdf/Pages%20from%20PK_Efremova_A4.pdf (дата обращения 25.05.2026)

5. Шеметова Е. Г., Шуршалова С. А., Усманова А. Е. К вопросу создания безопасных условий труда для пользователей персональных компьютеров в университете города Новосибирска // КАЧЕСТВО ПРОДУКЦИИ: КОНТРОЛЬ, УПРАВЛЕНИЕ, ПОВЫШЕНИЕ, ПЛАНИРОВАНИЕ. – 2016. – С. 354-357. [Электронный ресурс]. URL: <https://elibrary.ru/item.asp?id=27505685> (дата обращения 25.05.2026)

6. Колодешников А. С. РАССУЖДЕНИЕ О БЕЗОПАСНОСТИ МОЕЙ БУДУЩЕЙ ПРОФЕССИИ «СПЕЦИАЛИСТ ПО КОМПЬЮТЕРНЫМ СИСТЕМАМ» // Образование. Карьера. Общество. – 2024. – №. 2 (81). – С. 58-60. [Электронный ресурс] Режим доступа: <https://cyberleninka.ru/article/n/rassuzhdenie-o-bezopasnosti-moey-buduschey-professii-spetsialist-po-kompyuternym-sistemam> свободный. (дата обращения: 25.05.2026).

7. Кузнецов М. Г. Безопасность труда при работе за компьютером // Развитие АПК и сельских территорий в условиях модернизации экономики. – 2018. – С. 129-131. [Электронный ресурс] / Режим доступа: <https://elibrary.ru/item.asp?id=39285006> 8 свободный. (дата обращения: 10.05.2026)

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		62

Приложение А

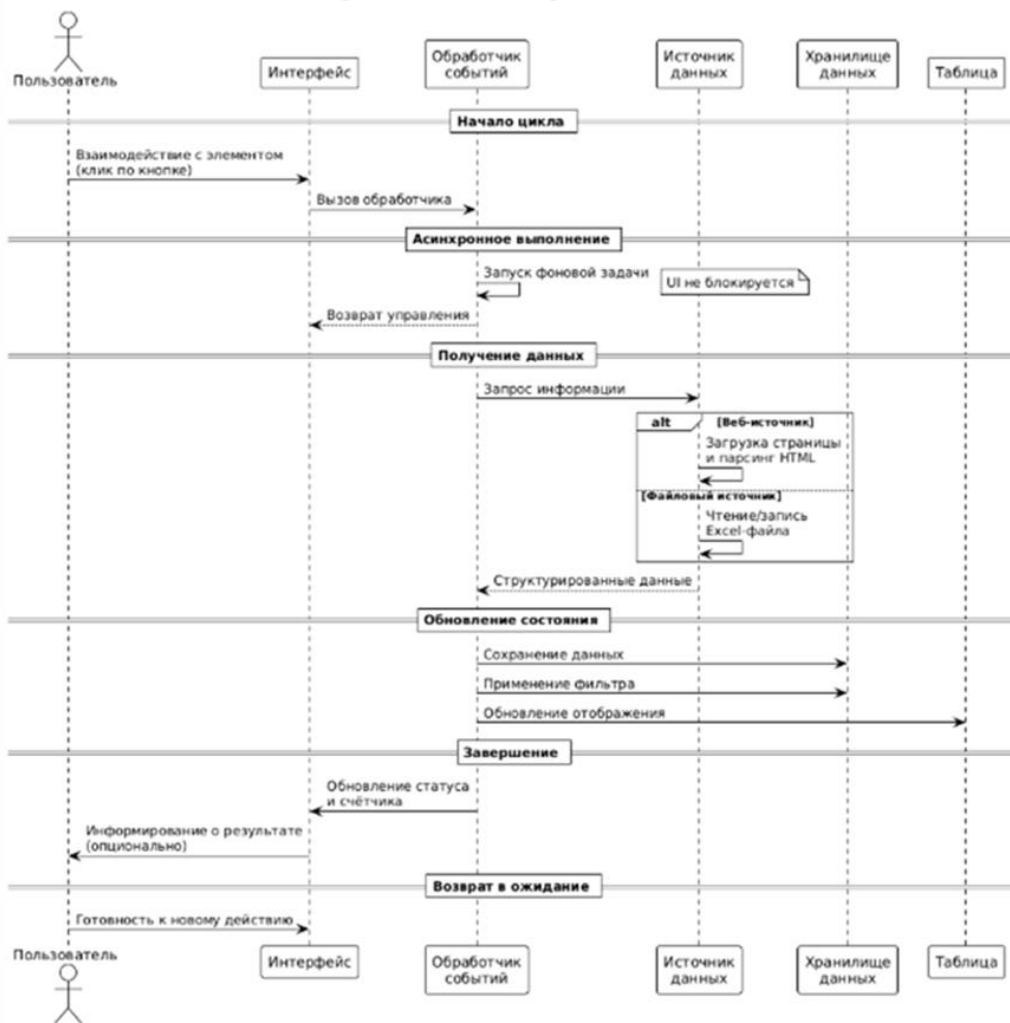
(обязательное)

«Программа составления отчётности майнинг-структуры»

UML диаграммы и схемы

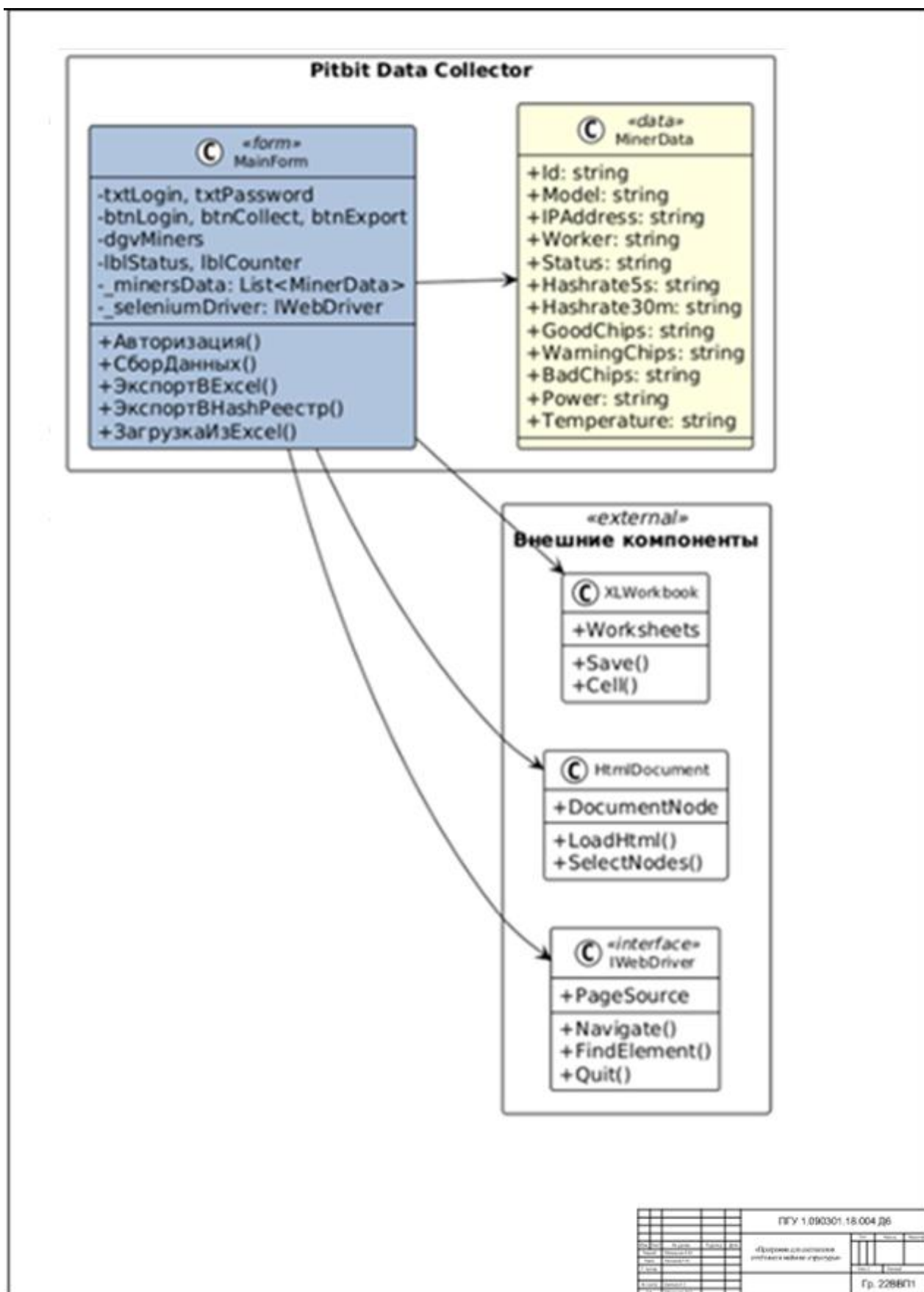
					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		63

Приложение А.1 — Функциональная схема работы проекта

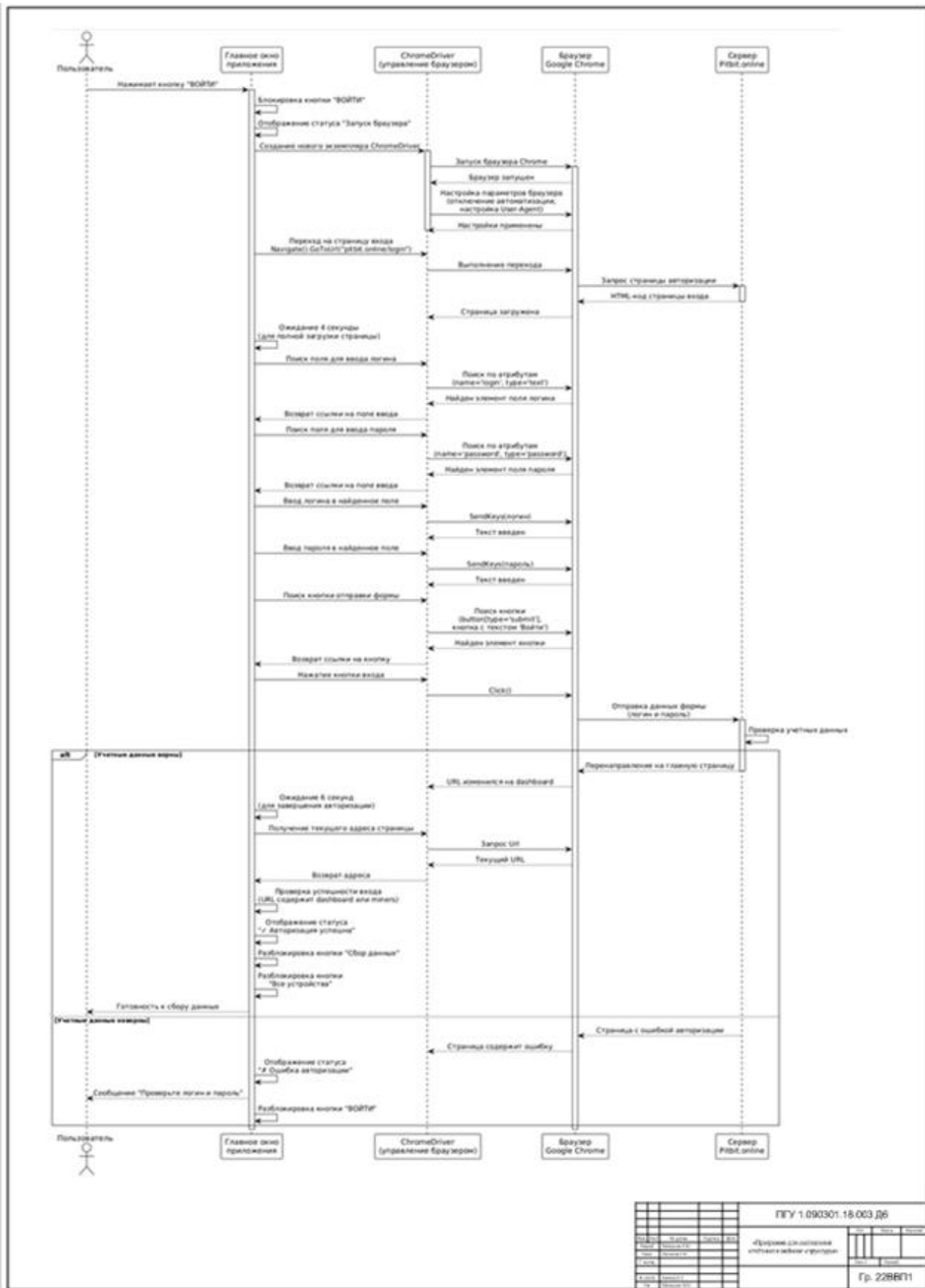


						ПГУ 1.090301.18.001 Д6			
Имя	Фамилия	Патроним	Дата	«Программа составления отчетности и мониторинг структур»			Имя	Фамилия	Патроним
Подпись	Место						Имя	Фамилия	Патроним
Дата	Подпись						Имя	Фамилия	Патроним
И. номер							Имя	Фамилия	Патроним
И. номер							Имя	Фамилия	Патроним
				Гр. 22ВВП1					

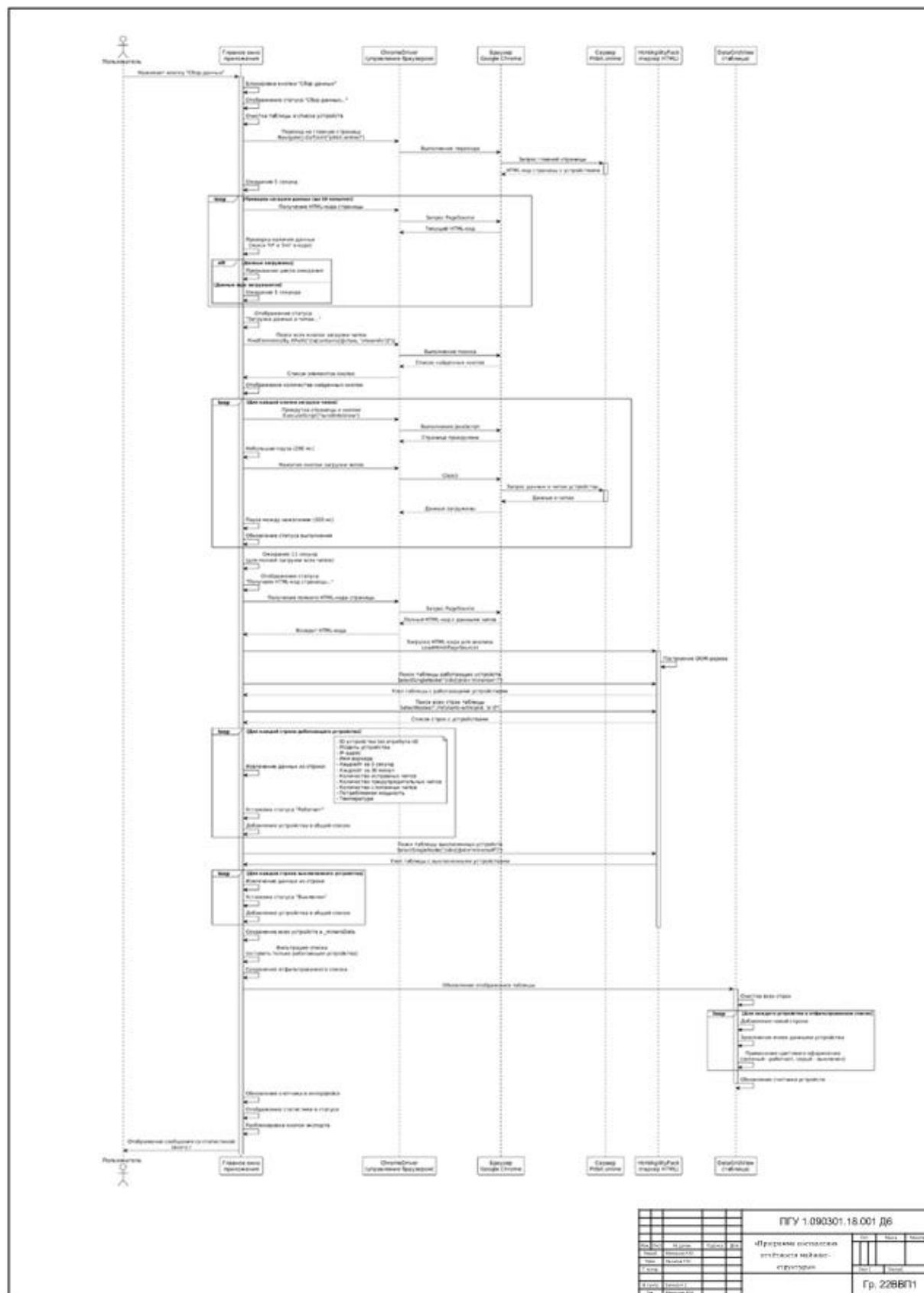
Приложение А.2 — Диаграмма классов

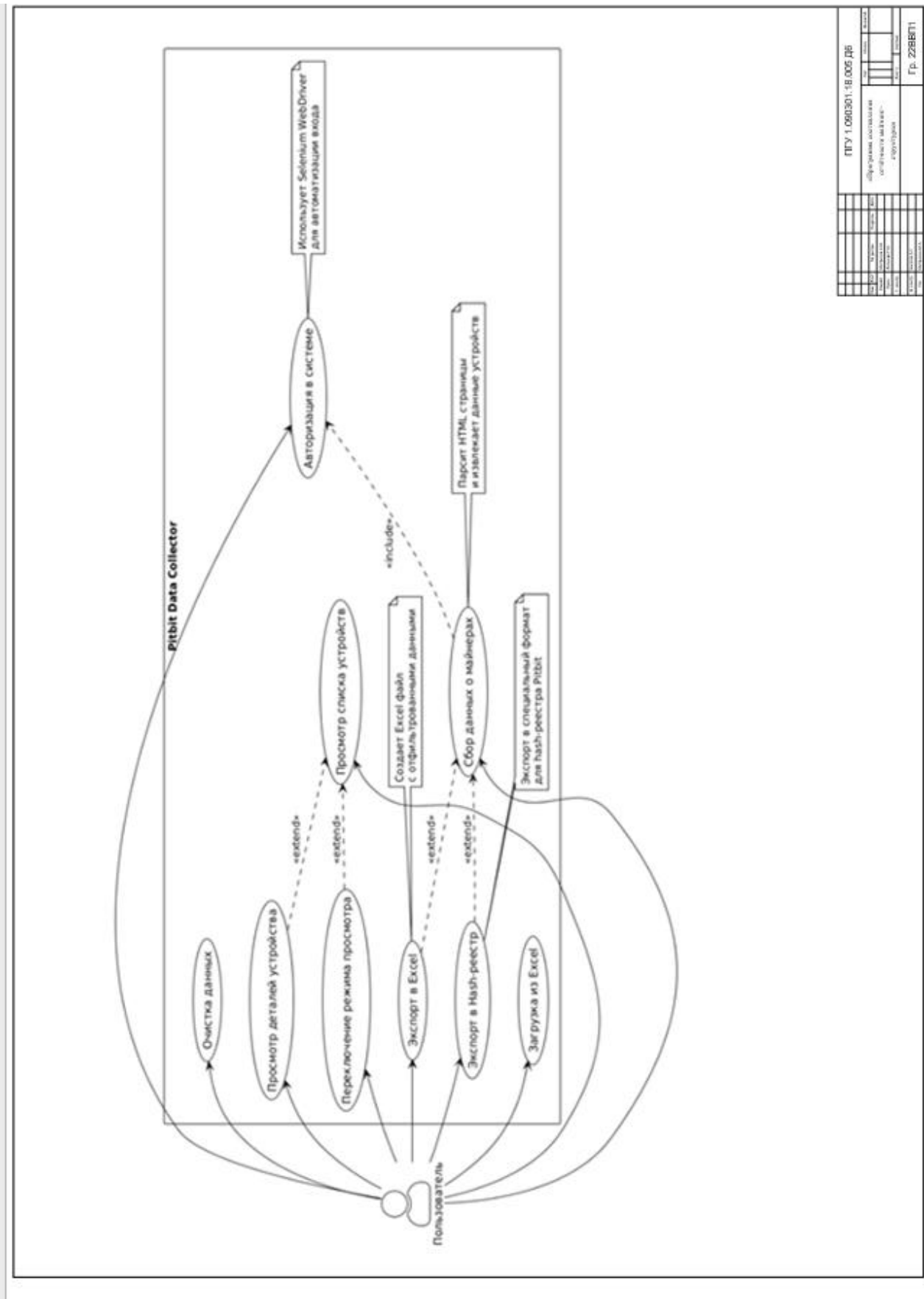


Приложение А.3 — Диаграмма последовательностей «Авторизация»



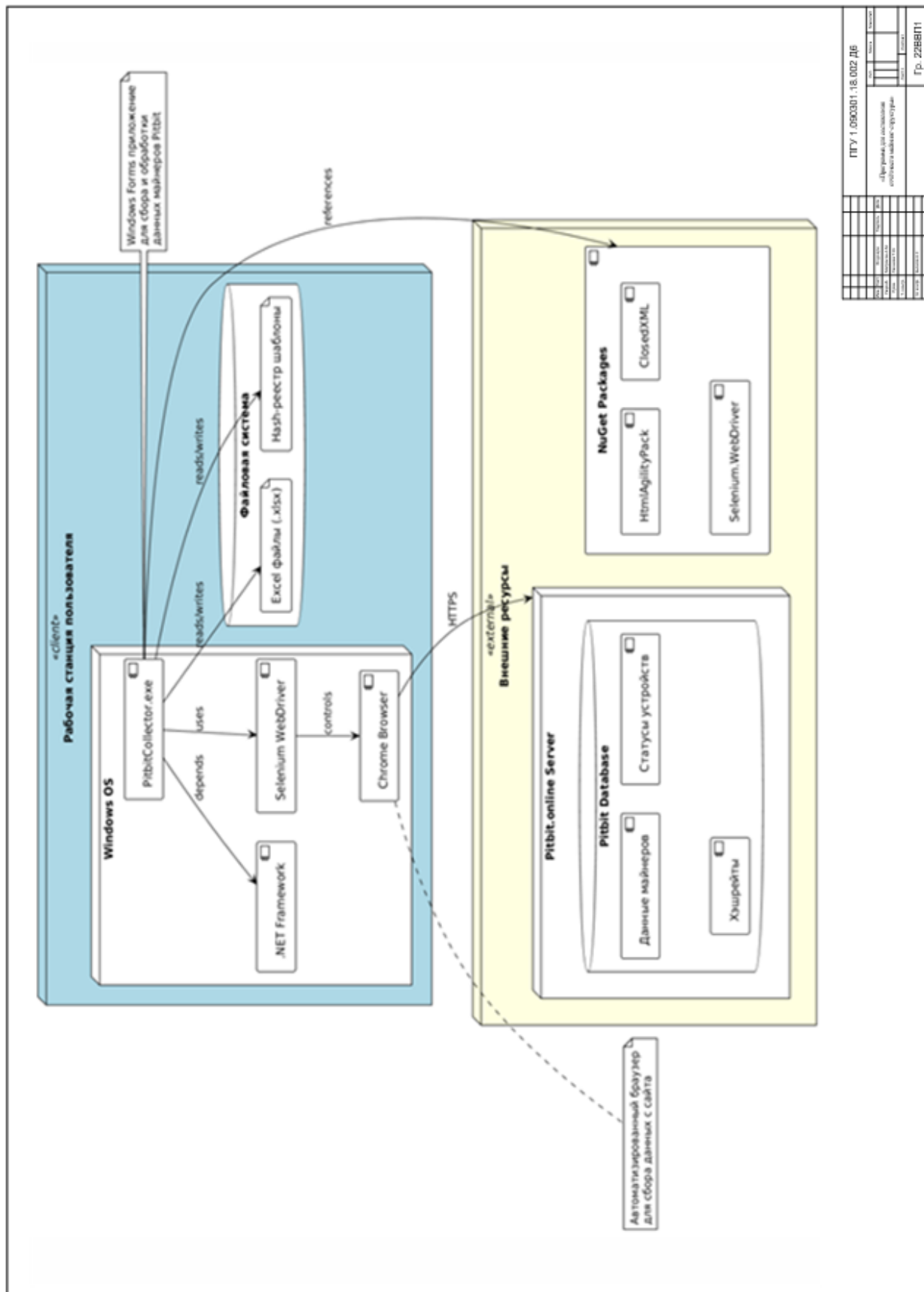
Приложение А.4 — Диаграмма последовательностей «Сбор данных»





ПГУ 1.090301.18.005 ДБ	
Имя	Иванов
Фамилия	Иванов
Пол	Мужской
Дата рождения	1990-01-01
Гр. 22ВВП1	

Приложение А.6 — Диаграмма развертывания

[illegible]

Приложение Б

(обязательное)

«Программа составления отчётности майнинг-структуры»

Листинг программ

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		70

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Net;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Text.RegularExpressions;
using HtmlAgilityPack;
using ClosedXML.Excel;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Linq;
using System.Reflection;

namespace PitbitCollector
{
    public class MainForm : Form
    {
        private Panel panel1;
        private Panel panel2;
        private TextBox txtLogin;
        private TextBox txtPassword;
        private Button btnLoginSelenium;
        private Label lblStatus;
        private DataGridView dgvMiners;
        private Button btnCollectData;
        private Button btnExportExcel;
        private Label label1;
        private Label label2;
        private Label label3;
        private Button btnClear;
        private Label lblCounter;
        private Button btnSwitchView;
        private Button btnExportHashRegistry;
        private Button btnLoadFromExcel;
        private Label lblTitle;
        private Label lblCounterLabel;

        private IWebDriver _seleniumDriver;
        private List<MinerData> _minersData;
        private List<MinerData> _currentDisplayedMiners;
        private bool _showingOnlineOnly = true;

        // Константы для пропорций
        private const int BUTTON_WIDTH = 150;
        private const int BUTTON_HEIGHT = 40;
        private const int BUTTON_MARGIN = 10;
        private const int PANEL_HEIGHT = 180;
        private const int TABLE_MARGIN_TOP = 250;

        public MainForm()

```

```

{
    InitializeComponents();
    _minersData = new List<MinerData>();
    _currentDisplayedMiners = new List<MinerData>();
    _seleniumDriver = null;
}

private void InitializeComponents()
{
    this.Text = "Pitbit Data Collector";
    this.Size = new Size(1200, 700);
    this.StartPosition = FormStartPosition.CenterScreen;
    this.BackColor = Color.FromArgb(207, 211, 219);
    this.MinimumSize = new Size(800, 500);
    this.Font = new Font("Segoe UI", 9);
    this.Resize += MainForm_Resize;

    // Главный заголовок
    lblTitle = new Label
    {
        Text = "PITBIT DATA COLLECTOR",
        Font = new Font("Segoe UI", 16, FontStyle.Bold),
        Location = new Point(20, 10),
        AutoSize = true,
        ForeColor = Color.FromArgb(102, 33, 102),
        Anchor = AnchorStyles.Top | AnchorStyles.Left
    };

    // Панель авторизации
    panell = new Panel
    {
        BackColor = Color.FromArgb(177, 94, 177),
        Location = new Point(20, 50),
        Size = new Size(350, PANEL_HEIGHT),
        BorderStyle = BorderStyle.FixedSingle,
        Anchor = AnchorStyles.Top | AnchorStyles.Left
    };

    label3 = new Label
    {
        Text = "ВХОД В АККАУНТ",
        Font = new Font("Segoe UI", 12, FontStyle.Bold),
        Location = new Point(110, 15),
        AutoSize = true,
        ForeColor = Color.White
    };

    label1 = new Label
    {
        Text = "Логин:",
        Font = new Font("Segoe UI", 10),
        Location = new Point(30, 60),
    }

```

```

        AutoSize = true,
        ForeColor = Color.White
    };

txtLogin = new TextBox
{
    Location = new Point(120, 55),
    Size = new Size(160, 30),
    Text = "+79603221527",
    Font = new Font("Segoe UI", 9),
    BackColor = Color.FromArgb(150, 37, 150),
    ForeColor = Color.White,
    BorderStyle = BorderStyle.FixedSingle,
    Anchor = AnchorStyles.Top | AnchorStyles.Left
};

label2 = new Label
{
    Text = "Пароль:",
    Font = new Font("Segoe UI", 10),
    Location = new Point(30, 100),
    AutoSize = true,
    ForeColor = Color.White
};

txtPassword = new TextBox
{
    Location = new Point(120, 95),
    Size = new Size(160, 30),
    PasswordChar = '*',
    Text = "25aa7a2210",
    Font = new Font("Segoe UI", 9),
    BackColor = Color.FromArgb(150, 37, 150),
    ForeColor = Color.White,
    BorderStyle = BorderStyle.FixedSingle,
    Anchor = AnchorStyles.Top | AnchorStyles.Left
};

// Кнопка ВОЙТИ
btnLoginSelenium = new Button
{
    Text = "ВОЙТИ",
    Font = new Font("Segoe UI", 10, FontStyle.Bold),
    Location = new Point(100, 135),
    Size = new Size(150, 35),
    BackColor = Color.FromArgb(150, 37, 150),
    ForeColor = Color.White,
    FlatStyle = FlatStyle.Flat,
    Anchor = AnchorStyles.Top | AnchorStyles.Left
};

btnLoginSelenium.Click += async (s, e) => await
LoginToPitbitSelenium();

```

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		73

```

        panel1.Controls.AddRange(new Control[] {
            label3, label1, txtLogin, label2, txtPassword,
btnLoginSelenium
        });

// Панель действий
panel2 = new Panel
{
    Location = new Point(400, 50),
    Size = new Size(760, PANEL_HEIGHT),
    BackColor = Color.Transparent,
    Anchor = AnchorStyles.Top | AnchorStyles.Left |
AnchorStyles.Right
};

// Кнопка Сбор данных
btnCollectData = new Button
{
    Text = "Сбор данных",
    Font = new Font("Segoe UI", 10, FontStyle.Bold),
    Location = new Point(0, 0),
    Size = new Size(BUTTON_WIDTH, BUTTON_HEIGHT),
    BackColor = Color.FromArgb(177, 94, 177),
    ForeColor = Color.White,
    FlatStyle = FlatStyle.Flat,
    Enabled = false,
    Anchor = AnchorStyles.Top | AnchorStyles.Left
};
btnCollectData.Click += async (s, e) => await
CollectMinerDataSelenium();

// Кнопка Загрузка из Excel
btnLoadFromExcel = new Button
{
    Text = "Загрузка из Excel",
    Font = new Font("Segoe UI", 10, FontStyle.Bold),
    Location = new Point(BUTTON_WIDTH + BUTTON_MARGIN,
0),

    Size = new Size(BUTTON_WIDTH, BUTTON_HEIGHT),
    BackColor = Color.FromArgb(177, 94, 177),
    ForeColor = Color.White,
    FlatStyle = FlatStyle.Flat,
    Anchor = AnchorStyles.Top | AnchorStyles.Left
};
btnLoadFromExcel.Click += btnLoadFromExcel_Click;

// Кнопка Все устройства
btnSwitchView = new Button
{
    Text = "Все устройства",
    Font = new Font("Segoe UI", 9, FontStyle.Bold),

```

```

        Location = new Point(2 * (BUTTON_WIDTH +
BUTTON_MARGIN), 0),
        Size = new Size(BUTTON_WIDTH, BUTTON_HEIGHT),
        BackColor = Color.FromArgb(177, 94, 177),
        ForeColor = Color.White,
        FlatStyle = FlatStyle.Flat,
        Enabled = false,
        Anchor = AnchorStyles.Top | AnchorStyles.Left
    };
    btnSwitchView.Click += btnSwitchView_Click;

    // Кнопка Экспорт в Excel
    btnExportExcel = new Button
    {
        Text = "Экспорт в Excel",
        Font = new Font("Segoe UI", 10, FontStyle.Bold),
        Location = new Point(3 * (BUTTON_WIDTH +
BUTTON_MARGIN), 0),
        Size = new Size(BUTTON_WIDTH, BUTTON_HEIGHT),
        BackColor = Color.FromArgb(177, 94, 177),
        ForeColor = Color.White,
        FlatStyle = FlatStyle.Flat,
        Anchor = AnchorStyles.Top | AnchorStyles.Left
    };
    btnExportExcel.Click += btnExportExcel_Click;

    // Кнопка Hash-реестр
    btnExportHashRegistry = new Button
    {
        Text = "Hash-реестр",
        Font = new Font("Segoe UI", 10, FontStyle.Bold),
        Location = new Point(0, BUTTON_HEIGHT +
BUTTON_MARGIN),
        Size = new Size(BUTTON_WIDTH, BUTTON_HEIGHT),
        BackColor = Color.FromArgb(177, 94, 177),
        ForeColor = Color.White,
        FlatStyle = FlatStyle.Flat,
        Enabled = false,
        Anchor = AnchorStyles.Top | AnchorStyles.Left
    };
    btnExportHashRegistry.Click +=
btnExportHashRegistry_Click;

    // Кнопка Очистить
    btnClear = new Button
    {
        Text = "Очистить",
        Font = new Font("Segoe UI", 10, FontStyle.Bold),
        Location = new Point(BUTTON_WIDTH + BUTTON_MARGIN,
BUTTON_HEIGHT + BUTTON_MARGIN),
        Size = new Size(BUTTON_WIDTH, BUTTON_HEIGHT),
        BackColor = Color.FromArgb(177, 94, 177),

```

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		75

```

        ForeColor = Color.White,
        FlatStyle = FlatStyle.Flat,
        Anchor = AnchorStyles.Top | AnchorStyles.Left
    };
    btnClear.Click += btnClear_Click;

    // Счетчик устройств
    lblCounterLabel = new Label
    {
        Text = "Количество устройств:",
        Font = new Font("Segoe UI", 10, FontStyle.Bold),
        Location = new Point(0, 110),
        AutoSize = true,
        ForeColor = Color.FromArgb(130, 9, 130),
        Anchor = AnchorStyles.Top | AnchorStyles.Left
    };

    lblCounter = new Label
    {
        Text = "0",
        Font = new Font("Segoe UI", 10, FontStyle.Bold),
        Location = new Point(180, 110),
        AutoSize = true,
        ForeColor = Color.Black,
        Anchor = AnchorStyles.Top | AnchorStyles.Left
    };

    // Статус программы
    lblStatus = new Label
    {
        Text = "Готов к работе",
        Font = new Font("Segoe UI", 10),
        Location = new Point(0, 140),
        AutoSize = true,
        ForeColor = Color.FromArgb(130, 9, 130),
        Anchor = AnchorStyles.Top | AnchorStyles.Left |
AnchorStyles.Right
    };

    panel2.Controls.AddRange(new Control[] {
        btnCollectData, btnLoadFromExcel, btnSwitchView,
        btnExportExcel, btnExportHashRegistry, btnClear,
        lblCounterLabel, lblCounter, lblStatus
    });

    // Таблица
    dgvMiners = new DataGridView
    {
        Location = new Point(20, TABLE_MARGIN_TOP),
        Size = new Size(1140, 400),
        AllowUserToAddRows = false,
        AllowUserToDeleteRows = false,

```

```

        ReadOnly = true,
        BackgroundColor = Color.White,
        BorderStyle = BorderStyle.Fixed3D,
        GridColor = Color.FromArgb(220, 220, 220),
        SelectionMode
DataGridViewSelectionMode.FullRowSelect,
        RowHeadersVisible = false,
        AutoSizeColumnsMode
DataGridViewAutoSizeColumnsMode.Fill,
        Font = new Font("Segoe UI", 9),
        RowTemplate = { Height = 30 },
        Anchor = AnchorStyles.Top | AnchorStyles.Bottom |
AnchorStyles.Left | AnchorStyles.Right
    };

    dgvMiners.Columns.Add("ID", "ID");
    dgvMiners.Columns.Add("Модель", "Модель");
    dgvMiners.Columns.Add("IP", "IP Адрес");
    dgvMiners.Columns.Add("Воркер", "Воркер");
    dgvMiners.Columns.Add("Хэшрейт 5с", "Хэшрейт 5с
(TH/s)");
    dgvMiners.Columns.Add("Хэшрейт 30мин", "Хэшрейт 30мин
(TH/s)");
    dgvMiners.Columns.Add("Чипы (хорошие)", "Чипы
(хорошие)");
    dgvMiners.Columns.Add("Чипы (плохие)", "Чипы
(плохие)");
    dgvMiners.Columns.Add("Чипы (слом.)", "Чипы
(слом.)");
    dgvMiners.Columns.Add("Всего чипов", "Всего чипов");
    dgvMiners.Columns.Add("Мощность", "Мощность (W)");
    dgvMiners.Columns.Add("Температура", "Температура
(°C)");
    dgvMiners.Columns.Add("Статус", "Статус");

    // Настройка минимальных ширин колонок
    dgvMiners.Columns["ID"].MinimumWidth = 50;
    dgvMiners.Columns["Модель"].MinimumWidth = 120;
    dgvMiners.Columns["IP"].MinimumWidth = 100;
    dgvMiners.Columns["Воркер"].MinimumWidth = 150;
    dgvMiners.Columns["Хэшрейт 5с"].MinimumWidth = 90;
    dgvMiners.Columns["Хэшрейт 30мин"].MinimumWidth = 90;
    dgvMiners.Columns["Чипы (хорошие)"].MinimumWidth =
80;
    dgvMiners.Columns["Чипы (плохие)"].MinimumWidth = 80;
    dgvMiners.Columns["Чипы (слом.)"].MinimumWidth = 80;
    dgvMiners.Columns["Всего чипов"].MinimumWidth = 80;
    dgvMiners.Columns["Мощность"].MinimumWidth = 90;
    dgvMiners.Columns["Температура"].MinimumWidth = 90;
    dgvMiners.Columns["Статус"].MinimumWidth = 80;

```

```

        dgvMiners.CellDoubleClick                                     +=
dgvMiners_CellDoubleClick;
        dgvMiners.CellFormatting += dgvMiners_CellFormatting;

        // Добавляем все контролы на форму
        this.Controls.AddRange(new Control[] {
            lblTitle,
            panel1,
            panel2,
            dgvMiners
        });

        // Первоначальная настройка размеров
        MainForm_Resize(this, EventArgs.Empty);
    }

    private void MainForm_Resize(object sender, EventArgs e)
    {
        int formWidth = this.ClientSize.Width;
        int formHeight = this.ClientSize.Height;

        // Панель действий растягивается по ширине
        panel2.Width = formWidth - panel2.Left - 20;
        panel2.Height = PANEL_HEIGHT;

        // Пересчитываем позиции кнопок
        int availableWidth = panel2.Width;
        int buttonsPerRow = Math.Max(1, availableWidth /
(BUTTON_WIDTH + BUTTON_MARGIN));

        if (buttonsPerRow > 6) buttonsPerRow = 6;

        // Первая строка кнопок
        var buttons =
panel2.Controls.OfType<Button>().ToList();
        for (int i = 0; i < Math.Min(4, buttons.Count); i++)
        {
            buttons[i].Location = new Point(i * (BUTTON_WIDTH
+ BUTTON_MARGIN), 0);
        }

        // Вторая строка кнопок
        int secondRowStartIndex = 4;
        for (int i = secondRowStartIndex; i <
Math.Min(secondRowStartIndex + 2, buttons.Count); i++)
        {
            int colIndex = i - secondRowStartIndex;
            buttons[i].Location = new Point(colIndex *
(BUTTON_WIDTH + BUTTON_MARGIN), BUTTON_HEIGHT + BUTTON_MARGIN);
        }

        // Позиция меток

```

```

        lblCounterLabel.Location = new Point(0, 2 *
(BUTTON_HEIGHT + BUTTON_MARGIN) + 10);
        lblCounter.Location = new Point(lblCounterLabel.Width
+ 10, lblCounterLabel.Top);
        lblStatus.Location = new Point(0,
lblCounterLabel.Bottom + 10);
        lblStatus.Width = panel2.Width - lblStatus.Left - 10;

        // Таблица растягивается
        dgvMiners.Width = formWidth - 40;
        dgvMiners.Height = formHeight - TABLE_MARGIN_TOP - 20;
    }

    private void dgvMiners_CellFormatting(object sender,
DataGridViewCellFormattingEventArgs e)
    {
        if (e.RowIndex < 0 || e.RowIndex >=
dgvMiners.Rows.Count) return;

        var columnName =
dgvMiners.Columns[e.ColumnIndex].Name;

        if (columnName == "Статус" && e.Value != null)
        {
            string status = e.Value.ToString();
            if (status == "Работает")
            {
                e.CellStyle.BackColor = Color.FromArgb(220,
255, 220);
                e.CellStyle.ForeColor = Color.DarkGreen;
                e.CellStyle.Font = new Font(dgvMiners.Font,
FontStyle.Bold);
            }
            else if (status == "Выключен")
            {
                e.CellStyle.BackColor = Color.LightGray;
                e.CellStyle.ForeColor = Color.DimGray;
            }
            else if (status == "Скрыт")
            {
                e.CellStyle.BackColor = Color.FromArgb(255,
240, 240);
                e.CellStyle.ForeColor = Color.DarkRed;
            }
        }

        if (columnName == "Чипы (хорошие)" && e.Value != null
&& e.Value.ToString() != "")
        {
            e.CellStyle.BackColor = Color.FromArgb(220, 255,
220);
            e.CellStyle.ForeColor = Color.DarkGreen;
        }
    }

```

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		79

```

        }
        else if (columnName == "Чипы (плохие)" && e.Value !=
null && e.Value.ToString() != "")
        {
            e.CellStyle.BackColor = Color.FromArgb(255, 255,
200);
            e.CellStyle.ForeColor = Color.OrangeRed;
        }
        else if (columnName == "Чипы (слом.)" && e.Value !=
null && e.Value.ToString() != "")
        {
            e.CellStyle.BackColor = Color.FromArgb(255, 220,
220);
            e.CellStyle.ForeColor = Color.DarkRed;
        }
    }

    private void btnSwitchView_Click(object sender, EventArgs
e)
    {
        if (_minersData.Count == 0)
        {
            MessageBox.Show("Нет данных для отображения",
"Информация",
                MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            return;
        }

        if (_showingOnlineOnly)
        {
            _currentDisplayedMiners = _minersData;
            _showingOnlineOnly = false;
            btnSwitchView.Text = "Только работающие";
            lblStatus.Text = $"Показаны все устройства:
{_minersData.Count}";
        }
        else
        {
            _currentDisplayedMiners = _minersData.Where(m =>
m.Status == "Работает").ToList();
            _showingOnlineOnly = true;
            btnSwitchView.Text = "Все устройства";
            lblStatus.Text = $"Показаны только работающие:
{_currentDisplayedMiners.Count}";
        }

        UpdateDataGridView(_currentDisplayedMiners);
    }

    private void UpdateDataGridView(List<MinerData>
minersToShow)

```

```

{
    dgvMiners.Rows.Clear();
    foreach (var miner in minersToShow)
    {
        int rowIndex = dgvMiners.Rows.Add();
        var dgvRow = dgvMiners.Rows[rowIndex];

        dgvRow.Cells[0].Value = miner.Id;
        dgvRow.Cells[1].Value = miner.Model;
        dgvRow.Cells[2].Value = miner.IPAddress;
        dgvRow.Cells[3].Value = miner.Worker;
        dgvRow.Cells[4].Value = miner.Hashrate5s;
        dgvRow.Cells[5].Value = miner.Hashrate30m;
        dgvRow.Cells[6].Value = miner.GoodChips;
        dgvRow.Cells[7].Value = miner.WarningChips;
        dgvRow.Cells[8].Value = miner.BadChips;
        dgvRow.Cells[9].Value = miner.TotalChips;
        dgvRow.Cells[10].Value = miner.Power;
        dgvRow.Cells[11].Value = miner.Temperature;
        dgvRow.Cells[12].Value = miner.Status;
    }
    lblCounter.Text = $"{minersToShow.Count}";
}

private void btnExportExcel_Click(object sender,
EventArgs e)
{
    if (_currentDisplayedMiners.Count == 0)
    {
        MessageBox.Show("Нет данных для экспорта!",
"Информация",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        return;
    }
    ExportToExcel();
}

private void btnExportHashRegistry_Click(object sender,
EventArgs e)
{
    if (_currentDisplayedMiners.Count == 0)
    {
        MessageBox.Show("Нет данных для экспорта в hash-
реестр!", "Информация",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        return;
    }
    ExportToHashRegistry();
}

```

```

        private void btnLoadFromExcel_Click(object sender,
EventArgs e)
        {
            LoadFromExcel();
        }

        private void btnClear_Click(object sender, EventArgs e)
        {
            _minersData.Clear();
            _currentDisplayedMiners.Clear();
            dgvMiners.Rows.Clear();
            lblStatus.Text = "Готов к работе";
            btnExportHashRegistry.Enabled = false;
            btnSwitchView.Enabled = false;
            btnSwitchView.Text = "Все устройства";
            _showingOnlineOnly = true;
            lblCounter.Text = "0";
        }

        private void dgvMiners_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
        {
            if (e.RowIndex >= 0 && e.RowIndex <
dgvMiners.Rows.Count)
            {
                string minerId =
dgvMiners.Rows[e.RowIndex].Cells[0].Value?.ToString();
                if (!string.IsNullOrEmpty(minerId))
                {
                    var miner = _minersData.Find(m => m.Id ==
minerId);
                    if (miner != null)
                    {
                        var details = $"ID: {miner.Id}\n" +
                                    $"Модель: {miner.Model}\n" +
                                    $"IP: {miner.IPAddress}\n" +
                                    $"Воркер: {miner.Worker}\n" +
                                    $"Хэшрейт 5с:
{miner.Hashrate5s} TH/s\n" +
                                    $"Хэшрейт 30мин:
{miner.Hashrate30m} TH/s\n" +
                                    $"Чипы хорошие:
{miner.GoodChips}\n" +
                                    $"Чипы плохие:
{miner.WarningChips}\n" +
                                    $"Чипы сломанные:
{miner.BadChips}\n" +
                                    $"Всего чипов:
{miner.TotalChips}\n" +
                                    $"Мощность: {miner.Power}\n"
+

```

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		82

```

        $"Температура:
{miner.Temperature}\n" +
        $"Статус: {miner.Status}";

        MessageBox.Show(details, "Информация о
майнере",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
}
}
private void LoadFromExcel()
{
    try
    {
        using (var openFileDialog = new OpenFileDialog())
        {
            openFileDialog.Filter = "Excel Files
(*.xlsx)|*.xlsx|All Files (*.*)|*.*";
            openFileDialog.Title = "Выберите файл с данными
майнеров";

            if (openFileDialog.ShowDialog() ==
DialogResult.OK)
            {
                lblStatus.Text = "Загрузка данных из
Excel...";

                using (var workbook = new
XLWorkbook(openFileDialog.FileName))
                {
                    var worksheet =
workbook.Worksheet(1);

                    _minersData.Clear();
                    _currentDisplayedMiners.Clear();

                    int row = 2;
                    while (!worksheet.Cell(row,
1).IsEmpty())
                    {
                        var miner = new MinerData
                        {
                            Id = worksheet.Cell(row,
1).GetValue<string>(),
                            Model = worksheet.Cell(row,
2).GetValue<string>(),
                            IPAddress =
worksheet.Cell(row, 3).GetValue<string>(),
                            Worker = worksheet.Cell(row,
4).GetValue<string>(),

```

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		83

```

                                Hashrate5s                =
worksheet.Cell(row, 5).GetValue<string>(),
                                Hashrate30m               =
worksheet.Cell(row, 6).GetValue<string>(),
                                GoodChips                  =
worksheet.Cell(row, 7).GetValue<string>(),
                                WarningChips               =
worksheet.Cell(row, 8).GetValue<string>(),
                                BadChips                   =
worksheet.Cell(row, 9).GetValue<string>(),
                                TotalChips                 =
worksheet.Cell(row, 10).GetValue<string>(),
                                Power   = worksheet.Cell(row,
11).GetValue<string>(),
                                Temperature               =
worksheet.Cell(row, 12).GetValue<string>(),
                                Status   = worksheet.Cell(row,
13).GetValue<string>()

                                };

                                _minersData.Add(miner);
                                row++;
                                }

                                _currentDisplayedMiners      =
_minersData.Where(m => m.Status == "Работает").ToList();
                                _showingOnlineOnly = true;
                                btnSwitchView.Text          =      "Все
устройства";

UpdateDataGridView(_currentDisplayedMiners);

                                int onlineCount = _minersData.Count(m
=> m.Status == "Работает");
                                int offlineCount          =
_minersData.Count(m => m.Status == "Выключен");
                                int hiddenCount = _minersData.Count(m
=> m.Status == "Скрыт");

                                lblCounter.Text            =
$" {_minersData.Count}";
                                lblStatus.Text = $"✓ Загружено из
Excel: { _minersData.Count} устройств";

                                btnExportHashRegistry.Enabled =
_currentDisplayedMiners.Count > 0;
                                btnSwitchView.Enabled        =
_minersData.Count > 0;

```

```

        MessageBox.Show($"Загружено из
Excel:\n\n" +
        $"Всего устройств:
{_minersData.Count}\n" +
        $"Работающих:
{onlineCount}\n" +
        $"Выключенных:
{offlineCount}\n" +
        $"Скрытых:
{hiddenCount}",
        "Загрузка завершена",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
}
catch (Exception ex)
{
    lblStatus.Text = $"X Ошибка загрузки:
{ex.Message}";
    MessageBox.Show($"Ошибка при загрузке из Excel:
{ex.Message}",
        "Ошибка",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}

private async Task LoginToPitbitSelenium()
{
    try
    {
        btnLoginSelenium.Enabled = false;
        btnLoginSelenium.Text = "Запуск...";
        lblStatus.Text = "Запуск браузера...";

        if (_seleniumDriver != null)
        {
            try { _seleniumDriver.Quit(); } catch { }
            _seleniumDriver = null;
        }

        var options = new ChromeOptions();
        options.AddArgument("--disable-blink-
features=AutomationControlled");
        options.AddExcludedArgument("enable-
automation");

        options.AddAdditionalOption("useAutomationExtension", false);
        options.AddArgument("user-agent=Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/124.0.0.0 Safari/537.36");
    }
}

```

```

options.AddArgument("--log-level=3");
options.AddArgument("--silent");

try
{
    _seleniumDriver = new ChromeDriver(options);
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка запуска
ChromeDriver: {ex.Message}\n\nУбедитесь, что установлен Chrome и
ChromeDriver.",
"Ошибка",
MessageBoxButtons.OK,
MessageBoxIcon.Error);
    throw;
}

_seleniumDriver.Manage().Timeouts().PageLoad =
    TimeSpan.FromSeconds(30);
_seleniumDriver.Manage().Window.Size = new
    System.Drawing.Size(1200, 800);

try
{
    _seleniumDriver.Navigate().GoToUrl("https://pitbit.online/login"
);

    await Task.Delay(4000);

    IWebElement loginField = null;
    IWebElement passwordField = null;
    IWebElement submitButton = null;

    var selectors = new[]
    {
        By.Name("login"),
        By.Name("phone"),
        By.Name("username"),
        By.CssSelector("input[type='text']"),
        By.CssSelector("input[type='tel']")
    };

    foreach (var selector in selectors)
    {
        try
        {
            var elements =
                _seleniumDriver.FindElements(selector);
            if (elements.Count > 0)
            {
                loginField = elements[0];
                break;
            }
        }
        catch { }
    }
}

```

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		86

```

        }
    }
    catch { }
}

var passwordSelectors = new[]
{
    By.Name("password"),
    By.CssSelector("input[type='password']")
};

foreach (var selector in passwordSelectors)
{
    try
    {
        var elements =
_seleniumDriver.FindElements(selector);
        if (elements.Count > 0)
        {
            passwordField = elements[0];
            break;
        }
    }
    catch { }
}

if (loginField == null || passwordField ==
null)
{
    throw new Exception("Не найдены поля для
ввода логина/пароля");
}

loginField.Clear();
loginField.SendKeys(txtLogin.Text);

passwordField.Clear();
passwordField.SendKeys(txtPassword.Text);

var buttonSelectors = new[]
{
    By.CssSelector("button[type='submit']"),
    By.XPath("//button[contains(text(),
'Вход') or contains(text(), 'Войти')]").
    By.CssSelector("input[type='submit']")
};

foreach (var selector in buttonSelectors)
{
    try
    {

```

```

        var elements =
_seleniumDriver.FindElements(selector);
        if (elements.Count > 0)
        {
            submitButton = elements[0];
            break;
        }
    }
    catch { }

    if (submitButton != null)
    {
        submitButton.Click();
    }
    else
    {

passwordField.SendKeys (OpenQA.Selenium.Keys.Enter);
    }

    await Task.Delay(6000);

    string currentUrl =
_seleniumDriver.Url.ToLower();
    string pageSource =
_seleniumDriver.PageSource;

    if (currentUrl.Contains("dashboard") ||
        currentUrl.Contains("miners") ||
        pageSource.Contains("minerson") ||
        !pageSource.Contains("sign in") ||
        pageSource.Contains("выход") ||
        pageSource.Contains("logout"))
    {
        lblStatus.Text = "✓ Авторизация успешна";
        btnCollectData.Enabled = true;
        btnSwitchView.Enabled = true;
    }
    else
    {
        lblStatus.Text = "✗ Ошибка авторизации";
        MessageBox.Show("Не удалось
авторизоваться. Проверьте логин и пароль.",
"Ошибка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}
catch (Exception ex)
{
    lblStatus.Text = $"✗ Ошибка: {ex.Message}";
}

```

```

        MessageBox.Show($"Ошибка при авторизации:
{ex.Message}",
        "Ошибка",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
catch (Exception ex)
{
    lblStatus.Text = $"X Ошибка: {ex.Message}";
}
finally
{
    btnLoginSelenium.Enabled = true;
    btnLoginSelenium.Text = "ВОЙТИ";
}
}

private async Task CollectMinerDataSelenium()
{
    try
    {
        if (_seleniumDriver == null)
        {
            MessageBox.Show("Сначала выполните вход",
            "Ошибка",
            MessageBoxButtons.OK,
            MessageBoxIcon.Warning);
            return;
        }

        btnCollectData.Enabled = false;
        btnExportHashRegistry.Enabled = false;
        lblStatus.Text = "Сбор данных...";

        dgvMiners.Rows.Clear();
        _minersData.Clear();
        _currentDisplayedMiners.Clear();

        _seleniumDriver.Navigate().GoToUrl("https://pitbit.online/");
        await Task.Delay(5000);

        int attempts = 0;
        while (attempts < 10)
        {
            string pageSource =
            _seleniumDriver.PageSource;
            if (pageSource.Contains("hf") &&
            pageSource.Contains("hm"))
            {
                break;
            }
        }
    }
}

```

```

        await Task.Delay(1000);
        attempts++;
    }

    lblStatus.Text = "Загрузка данных о чипах...";

    var chipButtons =
        _seleniumDriver.FindElements(By.XPath("//a[contains(@class,
        'showinfo') and contains(@addi, 'chips')]"));

    if (chipButtons.Count == 0)
    {
        chipButtons =
            _seleniumDriver.FindElements(By.CssSelector("a.showinfo[addi*='c
            hips']"));
    }

    if (chipButtons.Count == 0)
    {
        chipButtons =
            _seleniumDriver.FindElements(By.XPath("//a[contains(text(),
            'chips')]"));
    }

    int chipsButtonsClicked = 0;
    if (chipButtons.Count > 0)
    {
        lblStatus.Text = $"Найдено
        {chipButtons.Count} кнопок для загрузки чипов. Нажимаем...";

        foreach (var button in chipButtons)
        {
            try
            {
                ((IJavaScriptExecutor)_seleniumDriver).ExecuteScript("arguments[
                0].scrollIntoView(true);", button);
                await Task.Delay(200);

                button.Click();
                chipsButtonsClicked++;
                await Task.Delay(300);

                lblStatus.Text = $"Загружаем чипы";
                Application.DoEvents();
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Ошибка при
                нажатии кнопки чипов: {ex.Message}");
            }
        }
    }

```

```

        await Task.Delay(11000);

        lblStatus.Text = $"Нажато
{chipsButtonsClicked} кнопок. Получаем HTML...";
    }
    else
    {
        lblStatus.Text = "Кнопки загрузки чипов не
найденны. Пробуем получить данные без них...";
    }

    string pageContent = _seleniumDriver.PageSource;

    var htmlDoc = new HtmlAgilityPack.HtmlDocument();
    htmlDoc.LoadHtml(pageContent);

    var allMiners = new List<MinerData>();

    var onlineTable =
htmlDoc.DocumentNode.SelectSingleNode("//div[@id='minerson']");
    if (onlineTable != null)
    {
        var onlineRows =
onlineTable.SelectNodes(".//tr[starts-with(@id, 'a')]");
        if (onlineRows != null)
        {
            foreach (var row in onlineRows)
            {
                var miner = ParseMinerData(row);
                if (miner != null)
                {
                    miner.Status = "Работает";
                    allMiners.Add(miner);
                }
            }
        }
    }

    var offlineTable =
htmlDoc.DocumentNode.SelectSingleNode("//div[@id='minersoff']");
    if (offlineTable != null)
    {
        var offlineRows =
offlineTable.SelectNodes(".//tr[starts-with(@id, 'a')]");
        if (offlineRows != null)
        {
            foreach (var row in offlineRows)
            {
                var miner = ParseMinerData(row);
                if (miner != null)
                {

```

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.18.001 ПЗ

Лист

91

```

        miner.Status = "Выключен";
        allMiners.Add(miner);
    }
}

}

if (allMiners.Count == 0)
{
    var minerRows =
htmlDoc.DocumentNode.SelectNodes("//tr[starts-with(@id, 'a')]");

    if (minerRows != null && minerRows.Count > 0)
    {
        foreach (var row in minerRows)
        {
            var miner = ParseMinerData(row);
            if (miner != null)
            {
                string rowClass =
row.GetAttributeValue("class", "");
                if (rowClass.Contains("activen"))
                    miner.Status = "Работает";
                else if (rowClass.Contains("d-
none"))
                    miner.Status = "Скрыт";
                else
                    miner.Status = "Выключен";

                allMiners.Add(miner);
            }
        }
    }

    if (allMiners.Count == 0)
    {
        lblStatus.Text = "Устройства не найдены в
HTML";

        MessageBox.Show("Не найдены строки с
устройствами. Возможно, структура страницы изменилась.",
            "Ошибка",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        return;
    }

    _minersData = allMiners;

    _currentDisplayedMiners = _minersData.Where(m =>
m.Status == "Работает").ToList();
    _showingOnlineOnly = true;

```

```

        btnSwitchView.Text = "Все устройства";

        UpdateDataGridView(_currentDisplayedMiners);

        int onlineCount = _minersData.Count(m => m.Status
== "Работает");
        int offlineCount = _minersData.Count(m => m.Status
== "Выключен");
        int hiddenCount = _minersData.Count(m => m.Status
== "Скрыт");

        lblCounter.Text = $"{_minersData.Count}";
        lblStatus.Text = $"✓ Собрано данных:
{_minersData.Count} устройств,           показано
{_currentDisplayedMiners.Count} работающих";

        btnExportHashRegistry.Enabled =
        _currentDisplayedMiners.Count > 0;
        btnSwitchView.Enabled = _minersData.Count > 0;

        MessageBox.Show($"Собрано данных:\n\n" +
                        $"Всего           устройств:
{_minersData.Count}\n" +
                        $"Работающих: {onlineCount}\n" +
                        $"Выключенных: {offlineCount}\n" +
                        $"Скрытых: {hiddenCount}\n\n" +
                        $"В           таблице           показано:
{_currentDisplayedMiners.Count} работающих устройств",
                        "Статистика",           MessageBoxButtons.OK,
                        MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        lblStatus.Text = $"X Ошибка: {ex.Message}";
        MessageBox.Show($"Ошибка           сбора           данных:
{ex.Message}",
                        "Ошибка",           MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
    }
    finally
    {
        btnCollectData.Enabled = true;
    }
}

private MinerData ParseMinerData(HtmlNode row)
{
    try
    {
        var miner = new MinerData();

```

```

        string rowId = row.GetAttributeValue("id", "");
        if (rowId.StartsWith("a"))
        {
            miner.Id = rowId.Substring(1);
        }
        else
        {
            return null;
        }

        var modelLink =
row.SelectSingleNode(".//a[contains(@class, 'tdv')]");
        if (modelLink != null)
        {
            miner.Model =
CleanHtmlText(modelLink.InnerText);
        }

        var ipLink =
row.SelectSingleNode(".//a[contains(@href, '192.168.')]");
        if (ipLink != null)
        {
            miner.IPAddress = ipLink.InnerText.Trim();
        }

        var workerCell =
row.SelectSingleNode(".//td[contains(@class, 'pools') and @data-
bysort]");
        if (workerCell != null)
        {
            miner.Worker =
CleanHtmlText(workerCell.InnerText);
        }

        var hash5sElement =
row.SelectSingleNode($"//span[@id='hf{miner.Id}']");
        if (hash5sElement != null)
        {
            miner.Hashrate5s =
hash5sElement.InnerText.Trim();
        }

        var hash30mElement =
row.SelectSingleNode($"//span[@id='hm{miner.Id}']");
        if (hash30mElement != null)
        {
            miner.Hashrate30m =
hash30mElement.InnerText.Trim();
        }

        miner.GoodChips = "0";
        miner.WarningChips = "0";

```

```

miner.BadChips = "0";

var chipsCellWithData =
row.SelectSingleNode($"//td[contains(@class, 'chips') and
//span[contains(@id, 'cg{miner.Id}')] and string-
length(normalize-space(.)) > 0]");

if (chipsCellWithData == null)
{
    chipsCellWithData =
row.SelectSingleNode($"//td[contains(@class, 'chips') and @data-
bysort]");
}

if (chipsCellWithData != null)
{
    string chipsHtml =
chipsCellWithData.OuterHtml;

    var goodMatch = Regex.Match(chipsHtml,
@"cg{miner.Id}""?[^>]*>([<]+)");
    var warningMatch = Regex.Match(chipsHtml,
@"cy{miner.Id}""?[^>]*>([<]+)");
    var badMatch = Regex.Match(chipsHtml,
@"cr{miner.Id}""?[^>]*>([<]+)");

    if (goodMatch.Success)
    {
        miner.GoodChips =
ParseChipValue(goodMatch.Groups[1].Value.Trim());
    }

    if (warningMatch.Success)
    {
        miner.WarningChips =
ParseChipValue(warningMatch.Groups[1].Value.Trim());
    }

    if (badMatch.Success)
    {
        miner.BadChips =
ParseChipValue(badMatch.Groups[1].Value.Trim());
    }

    if (miner.GoodChips == "0")
    {
        var goodElement =
chipsCellWithData.SelectSingleNode($"//span[@id='cg{miner.Id}']
");
        if (goodElement != null)
        {

```

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		95

```

miner.GoodChips =
ParseChipValue(goodElement.InnerText.Trim());
    }
}

if (miner.WarningChips == "0")
{
    var warningElement =
chipsCellWithData.SelectSingleNode($"../span[@id='cy{miner.Id}']
");
    if (warningElement != null)
    {
        miner.WarningChips =
ParseChipValue(warningElement.InnerText.Trim());
    }
}

if (miner.BadChips == "0")
{
    var badElement =
chipsCellWithData.SelectSingleNode($"../span[@id='cr{miner.Id}']
");
    if (badElement != null)
    {
        miner.BadChips =
ParseChipValue(badElement.InnerText.Trim());
    }
}

}

var powerElement =
row.SelectSingleNode($"../*[@id='power{miner.Id}']");
if (powerElement != null)
{
    miner.Power = powerElement.InnerText.Trim();
}

var tempElement =
row.SelectSingleNode($"../*[@id='maxtemp{miner.Id}']");
if (tempElement != null)
{
    miner.Temperature =
ExtractTemperature(tempElement);
}

int good = string.IsNullOrEmpty(miner.GoodChips)
? 0 : int.TryParse(miner.GoodChips, out int g) ? g : 0;
int warning =
string.IsNullOrEmpty(miner.WarningChips) ? 0 :
int.TryParse(miner.WarningChips, out int w) ? w : 0;
int bad = string.IsNullOrEmpty(miner.BadChips) ?
0 : int.TryParse(miner.BadChips, out int b) ? b : 0;

```

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		96

```

        miner.TotalChips = (good + warning +
bad).ToString();

        return miner;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Ошибка парсинга строки:
{ex.Message}");
        return null;
    }
}

private string ParseChipValue(string text)
{
    if (string.IsNullOrEmpty(text)) return "0";

    string cleaned = Regex.Replace(text, @"^[^d]", "");

    if (!string.IsNullOrEmpty(cleaned))
        return cleaned;

    return "0";
}

private string ExtractTemperature(HtmlNode tempNode)
{
    if (tempNode == null) return "";

    var nextNode = tempNode.NextSibling;
    while (nextNode != null)
    {
        if (nextNode.NodeType == HtmlNodeType.Text &&
!string.IsNullOrEmpty(nextNode.InnerText))
        {
            return nextNode.InnerText.Trim();
        }
        nextNode = nextNode.NextSibling;
    }

    string temp = tempNode.InnerText.Trim();
    temp = Regex.Replace(temp, @"^[^d\.\-]", "").Trim();

    return temp;
}

private string CleanHtmlText(string htmlText)
{
    if (string.IsNullOrEmpty(htmlText))
        return "";

    string text = WebUtility.HtmlDecode(htmlText);

```

```

        text = Regex.Replace(text, @"<[^>]*>", "");
        text = Regex.Replace(text, @"\s+", " ").Trim();
        text = text.Replace("<v2>", "").Replace("</v2>",
        "").Trim();

        return text;
    }

    private void ExportToExcel()
    {
        try
        {
            using (var saveDialog = new SaveFileDialog())
            {
                saveDialog.Filter = "Excel Files
                (*.xlsx)|*.xlsx";

                string fileType = _showingOnlineOnly ?
                "работающие" : "все";
                saveDialog.FileName =
                $"pitbit_{fileType}_устройства_{DateTime.Now:yyyyMMdd_HH:mm:ss}.xl
                sx";

                if (saveDialog.ShowDialog() ==
                DialogResult.OK)
                {
                    using (var workbook = new XLWorkbook())
                    {
                        var worksheet =
                        workbook.Worksheets.Add("Майнеры");

                        string[] headers = {
                        "ID", "Модель", "IP Адрес", "Воркер",
                        "Хэшрейт 5с (TH/s)", "Хэшрейт 30мин
                        (TH/s)",
                        "Чипы хорошие", "Чипы плохие", "Чипы
                        сломанные", "Всего чипов",
                        "Мощность (W)", "Температура (°C)",
                        "Статус"
                        };

                        for (int i = 0; i < headers.Length;
                        i++)
                        {
                            var cell = worksheet.Cell(1, i +
                            1);

                            cell.Value = headers[i];
                            cell.Style.Font.Bold = true;
                            cell.Style.Fill.BackgroundColor =
                            XLColor.LightGray;
                        }
                    }
                }
            }
        }
    }

```

					ПГУ1.090301.18.001 ПЗ	Лист
						98
Изм.	Лист	№ докум.	Подпись	Дата		

```

                                for (int i = 0; i <
_currentDisplayedMiners.Count; i++)
                                {
                                    var miner =
_currentDisplayedMiners[i];
                                    int row = i + 2;

                                    worksheet.Cell(row, 1).Value =
miner.Id;
                                    worksheet.Cell(row, 2).Value =
miner.Model;
                                    worksheet.Cell(row, 3).Value =
miner.IPAddress;
                                    worksheet.Cell(row, 4).Value =
miner.Worker;
                                    worksheet.Cell(row, 5).Value =
miner.Hashrate5s;
                                    worksheet.Cell(row, 6).Value =
miner.Hashrate30m;
                                    worksheet.Cell(row, 7).Value =
miner.GoodChips;
                                    worksheet.Cell(row, 8).Value =
miner.WarningChips;
                                    worksheet.Cell(row, 9).Value =
miner.BadChips;
                                    worksheet.Cell(row, 10).Value =
miner.TotalChips;
                                    worksheet.Cell(row, 11).Value =
miner.Power;
                                    worksheet.Cell(row, 12).Value =
miner.Temperature;
                                    worksheet.Cell(row, 13).Value =
miner.Status;

                                    if (miner.Status == "Работает")
                                    {
                                        worksheet.Row(row).Style.Fill.BackgroundColor =
                                        XLColor.LightGreen;
                                    }
                                    else if (miner.Status ==
"Выключен")
                                    {
                                        worksheet.Row(row).Style.Fill.BackgroundColor =
                                        XLColor.LightGray;
                                    }
                                    else if (miner.Status == "Скрыт")
                                    {
                                        worksheet.Row(row).Style.Fill.BackgroundColor =
                                        XLColor.LightCoral;
                                    }
                                }

```

```

    }
}

worksheet.Columns().AdjustToContents();

workbook.SaveAs(saveDialog.FileName);
}

        lblStatus.Text = $"✓ Экспорт завершен:
{_currentDisplayedMiners.Count} устройств ({(_showingOnlineOnly ?
"только работающие" : "все устройства"))}";

        MessageBox.Show($"Файл                                успешно
сохранен:\n{saveDialog.FileName}\n\n" +
                        $"Экспортировано устройств:
{_currentDisplayedMiners.Count}",
                        "Экспорт                                завершен",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка                                при                                экспорте:
{ex.Message}", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}

private void ExportToHashRegistry()
{
    try
    {
        using (var openFileDialog = new OpenFileDialog())
        {
            openFileDialog.Filter = "Excel Files
(*.xlsx)|*.xlsx";
            openFileDialog.Title = "Выберите файл hash-
реестра";

            if (openFileDialog.ShowDialog() ==
                DialogResult.OK)
            {
                using (var workbook = new
                    XLWorkbook(openDialog.FileName))
                {
                    var sheetNames =
                        workbook.Worksheets.Select(ws => ws.Name).ToArray();

                    using (var sheetDialog = new Form())

```

					ПГУ1.090301.18.001 ПЗ	Лист
						100
Изм.	Лист	№ докум.	Подпись	Дата		

```

        {
            Text = "Выберите лист",
            Size = new Size(300, 200),
            StartPosition =
FormStartPosition.CenterParent
        })
        {
            var listBox = new ListBox()
            {
                Dock = DockStyle.Fill,
                DataSource = sheetNames
            };

            var btnOk = new Button()
            {
                Text = "OK",
                Dock = DockStyle.Bottom,
                DialogResult =
DialogResult.OK
            };

            sheetDialog.Controls.Add(listBox);
            sheetDialog.Controls.Add(btnOk);

            if (sheetDialog.ShowDialog() ==
DialogResult.OK && listBox.SelectedItem != null)
            {
                string selectedSheet =
listBox.SelectedItem.ToString();

                ExportToHashRegistryWorkbook(workbook, selectedSheet,
openDialog.FileName);
            }
        }
    }
}

catch (Exception ex)
{
    MessageBox.Show($"Ошибка при экспорте в hash-
реестр: {ex.Message}", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}

private void ExportToHashRegistryWorkbook(XLWorkbook
workbook, string sheetName, string filePath)
{
    try
    {

```

```

var worksheet = workbook.Worksheet(sheetName);

ClearExistingData(worksheet);

var minersRange1 = new List<MinerData>();
var minersRange2 = new List<MinerData>();

foreach (var miner in
_currentDisplayedMiners.Where(m => m.Status == "Работает"))
{
    string worker = miner.Worker?.ToLower() ?? "";

    if (worker.Contains("workerlesbx") ||
worker.Contains("workerlesax"))
    {
        minersRange1.Add(miner);
    }
    else if (worker.Contains("workerles00x"))
    {
        minersRange2.Add(miner);
    }
}

minersRange1 =
SortMinersByHashrate(minersRange1);
minersRange2 =
SortMinersByHashrate(minersRange2);

int s19Pro3x = 0, s19Pro2x = 0, s19Pro1x = 0;
int s19Exp2x = 0, s19Exp1x = 0;
int s21Hydro2x = 0, s21Hydro1x = 0;

int totalInserted = 0;

totalInserted += InsertMinersToRange(worksheet,
minersRange1, "B", "F", 8, 219);
totalInserted += InsertMinersToRange(worksheet,
minersRange2, "H", "L", 8, 52);

UpdateStatisticsForMiners(minersRange1.Concat(minersRange2).ToList(),
ref s19Pro3x, ref
s19Pro2x, ref s19Pro1x,
ref s19Exp2x, ref
s19Exp1x,
ref s21Hydro2x, ref
s21Hydro1x);

WriteStatistics(worksheet, s19Pro3x, s19Pro2x,
s19Pro1x,
s19Exp2x, s19Exp1x,

```

					ПГУ1.090301.18.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		102

```

        s21Hydro2x, s21Hydro1x);

        workbook.Save();

        int ignoredDevices = _currentDisplayedMiners
            .Where(m => m.Status == "Работает")
            .Count(m => {
                string worker = m.Worker?.ToLower() ?? "";
                return ! (worker.Contains("workerlesbx")
                    || worker.Contains("workerlesax") ||
                    worker.Contains("workerles00x"));
            });

        string sortingInfo = GetSortingInfo(minersRange1,
            minersRange2, ignoredDevices);

        lblStatus.Text = $"✓ Данные экспортированы в hash-
        реестр: {totalInserted} устройств";

        MessageBox.Show($"Экспорт в hash-реестр
        завершен!\n\n" +
            $"Вставлено устройств:
            {totalInserted}\n" +
            $"Диапазон B-F (workerlesbx/ax):
            {minersRange1.Count} устройств\n" +
            $"Диапазон H-L (workerles00x):
            {minersRange2.Count} устройств\n" +
            $"Проигнорировано других устройств:
            {ignoredDevices}\n\n" +
            $"{sortingInfo}\n\n" +
            $"S19 Pro 3x: {s19Pro3x}, 2x:
            {s19Pro2x}, 1x: {s19Pro1x}\n" +
            $"S19 EXP 2x: {s19Exp2x}, 1x:
            {s19Exp1x}\n" +
            $"S21 Hydro 2x: {s21Hydro2x}, 1x:
            {s21Hydro1x}\n\n" +
            $"Файл сохранен:
            {Path.GetFileName(filePath)}",
            "Экспорт завершен", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при записи в hash-
        реестр: {ex.Message}", "Ошибка",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

private void InsertDataAtRow(IXLWorksheet worksheet,
MinerData miner, string hash5s, string hash30m,
                        string startCol, int row)
{
    string numberFormat = "0";

    var ipCell = worksheet.Cell($"{startCol}{row}");
    ipCell.Value = miner.IPAddress;

    var modelCell = worksheet.Cell($"{GetNextColumn(startCol)}{row}");
    modelCell.Value = miner.Model;

    var hash5sCell = worksheet.Cell($"{GetNextColumn(startCol, 2)}{row}");
    SetNumericValue(hash5sCell, hash5s, numberFormat);

    var hash30mCell = worksheet.Cell($"{GetNextColumn(startCol, 3)}{row}");
    SetNumericValue(hash30mCell, hash30m, numberFormat);

    var chipsCell = worksheet.Cell($"{GetNextColumn(startCol, 4)}{row}");
    SetNumericValue(chipsCell, miner.GoodChips, numberFormat, true);
}

private void SetNumericValue(IXLCell cell, string value,
string format, bool isInteger = false)
{
    if (string.IsNullOrEmpty(value))
    {
        cell.Value = "";
        return;
    }

    string cleanValue = value.Trim()
        .Replace("TH/s", "")
        .Replace(" ", "")
        .Replace(",", ".");

    if (isInteger)
    {
        cleanValue = Regex.Replace(cleanValue, @"[^\\d]",
        "");

        if (int.TryParse(cleanValue, out int intValue))
        {
            cell.Value = intValue;
            cell.Style.NumberFormat.Format = format;
            cell.Style.Alignment.Horizontal =
            XLAAlignmentHorizontalValues.Right;

```

```

        }
        else
        {
            cell.Value = value;
        }
    }
    else
    {
        if (double.TryParse(cleanValue,
System.Globalization.NumberStyles.Any,
System.Globalization.CultureInfo.InvariantCulture, out double
doubleValue))
        {
            cell.Value = doubleValue;
            cell.Style.NumberFormat.Format = format;
            cell.Style.Alignment.Horizontal =
XLAAlignmentHorizontalValues.Right;
        }
        else
        {
            cell.Value = value;
        }
    }
}

private void ClearExistingData(IXLWorksheet worksheet)
{
    try
    {
        for (int row = 8; row <= 219; row++)
        {
            worksheet.Cell($"B{row}").Clear();
            worksheet.Cell($"C{row}").Clear();
            worksheet.Cell($"D{row}").Clear();
            worksheet.Cell($"E{row}").Clear();
            worksheet.Cell($"F{row}").Clear();
        }

        for (int row = 8; row <= 52; row++)
        {
            worksheet.Cell($"H{row}").Clear();
            worksheet.Cell($"I{row}").Clear();
            worksheet.Cell($"J{row}").Clear();
            worksheet.Cell($"K{row}").Clear();
            worksheet.Cell($"L{row}").Clear();
        }

        string[] statCells = { "I2", "J2", "K2", "O2",
"P2", "S2", "T2" };
        foreach (var cell in statCells)
        {

```

```

        worksheet.Cell(cell).Clear();
    }
}
catch (Exception ex)
{
    Console.WriteLine($"Ошибка при очистке данных:
{ex.Message}");
}

private string FormatHashrateWithoutTrailingZeros(string
hashrate)
{
    if (string.IsNullOrEmpty(hashrate)) return "0";

    if (hashrate.Contains("."))
    {
        hashrate = hashrate.TrimEnd('0').TrimEnd('.');
    }

    return hashrate;
}

private string ConvertHashrate(string hashrate)
{
    if (string.IsNullOrEmpty(hashrate)) return "0";

    string cleaned = hashrate.Replace("TH/s",
    "").Replace(" ", "").Trim();
    cleaned = cleaned.Replace(",", ".");

    if (double.TryParse(cleaned,
    System.Globalization.NumberStyles.Any,
    System.Globalization.CultureInfo.InvariantCulture, out double
    value))
    {
        double result = value * 1000;
        return
        result.ToString(System.Globalization.CultureInfo.InvariantCulture);
    }

    return "0";
}

private string GetNextColumn(string column, int steps = 1)
{
    for (int i = 0; i < steps; i++)
    {
        column = GetNextColumnSingle(column);
    }
}

```

```

        return column;
    }

    private string GetNextColumnSingle(string column)
    {
        char[] chars = column.ToCharArray();

        for (int i = chars.Length - 1; i >= 0; i--)
        {
            if (chars[i] < 'Z')
            {
                chars[i]++;
                break;
            }
            else
            {
                chars[i] = 'A';
                if (i == 0)
                {
                    return "A" + new string(chars);
                }
            }
        }

        return new string(chars);
    }

    private void WriteStatistics(IXLWorksheet worksheet,
                                int s19Pro3x, int s19Pro2x, int
s19Pro1x,
                                int s19Exp2x, int s19Exp1x,
                                int s21Hydro2x, int s21Hydro1x)
    {
        var cellI2 = worksheet.Cell("I2");
        cellI2.Value = s19Pro3x;
        cellI2.Style.NumberFormat.Format = "0";

        var cellJ2 = worksheet.Cell("J2");
        cellJ2.Value = s19Pro2x;
        cellJ2.Style.NumberFormat.Format = "0";

        var cellK2 = worksheet.Cell("K2");
        cellK2.Value = s19Pro1x;
        cellK2.Style.NumberFormat.Format = "0";

        var cellO2 = worksheet.Cell("O2");
        cellO2.Value = s19Exp2x;
        cellO2.Style.NumberFormat.Format = "0";

        var cellP2 = worksheet.Cell("P2");
        cellP2.Value = s19Exp1x;
        cellP2.Style.NumberFormat.Format = "0";
    }

```

```

        var cellS2 = worksheet.Cell("S2");
        cellS2.Value = s21Hydro2x;
        cellS2.Style.NumberFormat.Format = "0";

        var cellT2 = worksheet.Cell("T2");
        cellT2.Value = s21Hydro1x;
        cellT2.Style.NumberFormat.Format = "0";
    }

    private List<MinerData>
SortMinersByHashrate(List<MinerData> miners)
    {
        return miners.OrderByDescending(m => {
            if (!string.IsNullOrEmpty(m.Hashrate30m))
            {
                string cleanHash = m.Hashrate30m
                    .Replace("TH/s", "")
                    .Replace(" ", "")
                    .Replace(",", ".")
                    .Trim();

                if (double.TryParse(cleanHash,
System.Globalization.NumberStyles.Any,
System.Globalization.CultureInfo.InvariantCulture, out double
hashValue))
                {
                    return hashValue;
                }
            }
            return 0.0;
        }).ToList();
    }

    private int InsertMinersToRange(IXLWorksheet worksheet,
List<MinerData> miners,
string startCol, string
endCol, int startRow, int maxRow)
    {
        int insertedCount = 0;
        int currentRow = startRow;

        foreach (var miner in miners)
        {
            if (currentRow > maxRow)
            {
                MessageBox.Show($"Диапазон
{startCol}{startRow}:{endCol}{maxRow} заполнен. " +
                    $"Не все устройства
({miners.Count}) были вставлены.",

```

```

        "Предупреждение",    MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
        break;
    }

    string hash5s =
    ConvertHashrate(miner.Hashrate5s);
    string hash30m =
    ConvertHashrate(miner.Hashrate30m);

    InsertDataAtRow(worksheet,    miner,    hash5s,
    hash30m, startCol, currentRow);
    insertedCount++;
    currentRow++;
}

return insertedCount;
}

private void UpdateStatisticsForMiners(List<MinerData>
miners,
                                ref int s19Pro3x,
ref int s19Pro2x, ref int s19Pro1x,
                                ref int s19Exp2x,
ref int s19Exp1x,
                                ref int s21Hydro2x,
ref int s21Hydro1x)
{
    foreach (var miner in miners)
    {
        int goodChips = int.TryParse(miner.GoodChips, out
int g) ? g : 0;

        if (miner.Model.Contains("S19 Pro Hyd"))
        {
            if (goodChips > 360 && goodChips <= 540)
s19Pro3x++;
            else if (goodChips > 180 && goodChips <= 360)
s19Pro2x++;
            else if (goodChips > 0 && goodChips <= 180)
s19Pro1x++;
        }
        else if (miner.Model.Contains("S19 XP Hyd") ||
miner.Model.Contains("S19 EXP"))
        {
            if (goodChips > 204 && goodChips <= 408)
s19Exp2x++;
            else if (goodChips > 0 && goodChips <= 204)
s19Exp1x++;
        }
        else if (miner.Model.Contains("S21 Hydro"))
        {

```

```

        if (goodChips > 216 && goodChips <= 432)
s21Hydro2x++;
        else if (goodChips > 0 && goodChips <= 216)
s21Hydro1x++;
    }
}

private string GetSortingInfo(List<MinerData>
range1Miners, List<MinerData> range2Miners, int ignoredCount)
{
    string info = "Информация о сортировке:\n";

    if (range1Miners.Count > 0)
    {
        info += $"Диапазон В-Ф отсортирован:
{range1Miners.Count} устройств\n";
        if (range1Miners.Count >= 2)
        {
            var top1 = range1Miners[0];
            var last1 = range1Miners[range1Miners.Count -
1];
            info += $"Первое: {top1.Hashrate30m}
({top1.Model})\n";
            info += $"Последнее: {last1.Hashrate30m}
({last1.Model})\n";
        }
    }
    else
    {
        info += $"Диапазон В-Ф: нет подходящих
устройств\n";
    }

    if (range2Miners.Count > 0)
    {
        info += $"Диапазон Н-Л отсортирован:
{range2Miners.Count} устройств\n";
        if (range2Miners.Count >= 2)
        {
            var top2 = range2Miners[0];
            var last2 = range2Miners[range2Miners.Count -
1];
            info += $"Первое: {top2.Hashrate30m}
({top2.Model})\n";
            info += $"Последнее: {last2.Hashrate30m}
({last2.Model})\n";
        }
    }
    else
    {

```

```

        info += $"Диапазон H-L: нет подходящих
устройств\n";
    }

    if (ignoredCount > 0)
    {
        info += $" \n Другие устройства: {ignoredCount}\n";
        info += "(устройства с другими воркерами не
вставляются в hash-реестр)";
    }

    return info;
}

protected override void
OnFormClosing(FormClosingEventArgs e)
{
    base.OnFormClosing(e);

    if (_seleniumDriver != null)
    {
        try
        {
            _seleniumDriver.Quit();
            _seleniumDriver.Dispose();
        }
        catch { }
    }
}

public class MinerData
{
    public string Id { get; set; }
    public string Model { get; set; }
    public string IPAddress { get; set; }
    public string Worker { get; set; }
    public string Hashrate5s { get; set; }
    public string Hashrate30m { get; set; }
    public string GoodChips { get; set; }
    public string WarningChips { get; set; }
    public string BadChips { get; set; }
    public string TotalChips { get; set; }
    public string Power { get; set; }
    public string Temperature { get; set; }
    public string Status { get; set; }
}
}

```

Приложение В

«Программа составления отчётности майнинг-структуры» Презентация

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ПРОГРАММА СОСТАВЛЕНИЯ ОТЧЁТНОСТИ МАЙНИНГ- СТРУКТУРЫ

АВТОР РАБОТЫ: СТУДЕНТ ГР.22ВВП1 Матросов А.М.

НАУЧНЫЙ РУКОВОДИТЕЛЬ: ДОЦЕНТ, К.П.Н. Леонова Т.Ю.

Цель работы

- Разработка инструмента, автоматизирующего процесс сбора и первичной обработки телеметрии оборудования.

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.18.001 ПЗ

Лист

112

Актуальность

- Темы криптовалюты и майнинга становятся все более популярными в наше время. Одной из важнейших характеристик майнинг — системы является ее скорость. И именно отслеживание работы устройств позволяет определить, какие аспекты можно изменить в ходе работы самих машин для майнинга.



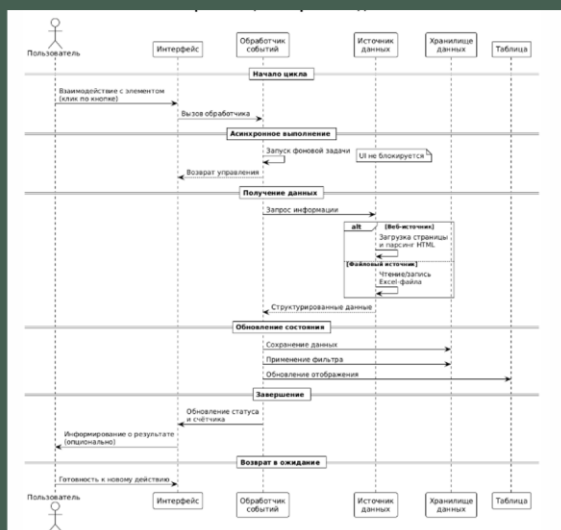
Функционал

- вход в личный кабинет платформы;
- автоматический сбор и парсинг данных;
- отображение собранных данных в табличном виде;
- экспорт данных в Excel-таблицы;
- импорт данных из Excel-таблицы;
- интуитивно-понятный интерфейс;
- отображение подробной информации.



Изм.	Лист	№ докум.	Подпись	Дата

Выбор архитектуры



Технологии разработки

Среда разработки



Язык разработки



Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.18.001 ПЗ

Лист

114

```

    usecaseDiagram
        actor User as Пользователь
        participant PitbitDataCollector as Pitbit Data Collector
        participant PitbitDataCollector as Pitbit Data Collector

        User --> UC1[Очистка данных]
        User --> UC2[Просмотр деталей устройства]
        User --> UC3[Переключение режима просмотра]
        User --> UC4[Экспорт в Excel]
        User --> UC5[Экспорт в Hash-реестр]
        User --> UC6[Загрузка из Excel]

        PitbitDataCollector --> UC1
        PitbitDataCollector --> UC2
        PitbitDataCollector --> UC3
        PitbitDataCollector --> UC4
        PitbitDataCollector --> UC5
        PitbitDataCollector --> UC6

        UC2 -.-> UC7[Просмотр списка устройств] : «extend»
        UC3 -.-> UC7 : «extend»
        UC4 -.-> UC7 : «extend»
        UC5 -.-> UC7 : «extend»
        UC6 -.-> UC7 : «extend»

        UC7 --> UC8[Авторизация в системе] : «include»
        UC8 --> UC9[Использует Selenium WebDriver для автоматизации ввода]

        UC7 --> UC10[Сбор данных о майнерах] : «include»
        UC10 --> UC11[Парсит HTML-страницы и извлекает данные устройств]

        UC7 --> UC12[Экспорт в специальный формат для hash-реестра Pitbit]
    
```

The diagram illustrates the interactions between a user and the Pitbit Data Collector system. The user can perform several actions: cleaning data, viewing device details, switching viewing modes, exporting to Excel, exporting to a Hash registry, and loading from Excel. The system (Pitbit Data Collector) handles these actions. The 'Просмотр списка устройств' (View list of devices) use case is extended by 'Просмотр деталей устройства', 'Переключение режима просмотра', 'Экспорт в Excel', 'Экспорт в Hash-реестр', and 'Загрузка из Excel'. This use case includes 'Авторизация в системе' (Authorization in the system) and 'Сбор данных о майнерах' (Data collection about miners). The 'Авторизация в системе' use case includes the note 'Использует Selenium WebDriver для автоматизации ввода' (Uses Selenium WebDriver for input automation). The 'Сбор данных о майнерах' use case includes the note 'Парсит HTML-страницы и извлекает данные устройств' (Parses HTML pages and extracts device data). Finally, the 'Просмотр списка устройств' use case includes the note 'Экспорт в специальный формат для hash-реестра Pitbit' (Export to a special format for the Pitbit hash registry).

The screenshot displays the 'PITBIT Data Collector' web application. The interface is divided into several sections:

- Header:** The title 'PITBIT Data Collector' is located in the top left corner.
- Login Section (Left):** A purple sidebar contains the text 'ВХОД В АККАУНТ' (Login to Account). Below it are input fields for 'Логин:' (Login) and 'Пароль:' (Password), followed by a 'ВОЙТИ' (Login) button.
- Navigation Buttons (Top Center):** A row of four buttons: 'Сбор данных' (Data Collection), 'Загрузка из Excel' (Load from Excel), 'Все устройства' (All Devices), and 'Экспорт в Excel' (Export to Excel).
- Management Buttons (Below Navigation):** Two buttons: 'Наш-реестр' (Our Registry) and 'Очистить' (Clear).
- Device Count (Below Management Buttons):** A label 'Количество устройств:' (Number of devices:) followed by the value '0' and a link 'Готов к работе' (Ready for work).
- Table:** A table with 12 columns: ID, Модель (Model), IP Адрес (IP Address), Выход (Exit), Частота 5с (THz) (Frequency 5s (THz)), Частота 10мин (Frequency 10min), Четы (Chety), Четы (плоские) (Chety (flat)), Четы (сломы) (Chety (broken)), Всего чипов (Total chips), Мощность (W) (Power (W)), and Температура (°C) (Temperature (°C)). The table is currently empty.