

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
МО и ПЭВМ

Направление 09.04.04 «Программная инженерия»
подготовки
Магистерская Программное обеспечение вычислительной
программа техники и автоматизированных систем

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему

**Исследование и разработка автоматизированных
обучающих систем с использованием технологии
дополненной реальности**

Студент АК 28.06.21 Акчурина Варвара Дмитриевна
(подпись, дата) (ФИО полностью)

Руководитель В.Н. 28.06.21 Князев В.Н.
(подпись, дата) (фамилия, инициалы)

Нормоконтролер Д.В. 28.06.21 Такташкин Д.В.
(подпись, дата) (фамилия, инициалы)

Рецензент Начальник отдела ООО А.Ю. 28.06.21 Афонин А.Ю.
(подпись, дата) (фамилия, инициалы)
НТП «Криптософт»
(должность, место работы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой П.П. 28.06.21 Макарычев П.П.
(подпись) (фамилия, инициалы)

Работа защищена с отметкой 5 (протокол заседания ГЭК от 28.06.21 № 2)

Секретарь ГЭК Н.А. 28.06.21 Попова Н.А.
(подпись) (фамилия, инициалы)

Пенза, 2021

Список сокращений

ПС	Программные средства
ПО	Программное обеспечение
АОС	Автоматизированная обучающая система
UML	Unified modeling language
RUP	Rational unified process
ООП	Объектно-ориентированное программирование
IT	Informational technologies
ВАК	Высшая аттестационная комиссия
MS	Microsoft

Содержание

Введение.....	5
1 Анализ предметной области исследования	9
1.1 Классификация автоматизированных обучающих систем	9
1.1.1 Современная классификация АОС.....	9
1.1.2 Расширенная классификация АОС.....	11
1.2 Анализ аналогичных автоматизированных обучающих систем	13
1.2.1 Компьютерный лабораторный практикум по физике	13
1.2.2 Компьютерный справочник с сайта all-fizika.com.....	14
1.2.3 Справочник по физике для смартфонов	16
1.2.4 Общие выводы по анализу аналогов.....	17
1.3 Модель освоения учебного материала.....	17
1.3.1 Общее описание модели освоения учебного материала	17
1.3.2 Расширенная модель освоения учебного материала	20
1.4 Планирование разработки и технико-экономическое обоснование	25
1.5 Выбор технологии разработки.....	36
1.6 Выбор программных средств	37
Выводы	38
2 Разработка алгоритма формирования расширенной модели	39
2.1 Онтологическое моделирование.....	39
2.2 Анализ требований.....	40
2.2.1 Анализ функциональных требований	40
2.2.2 Анализ нефункциональных требований	45
2.3 Разработка и исследование алгоритма.....	45
2.3.1 Аналитическое исследование с применением сетей Петри	45

2.3.2 Эксперимент в программе Petri net emulator	55
2.4 Разработка интерфейса приложения	59
2.4.1 Разработка интерфейса программы формирования модели	59
2.4.2 Разработка интерфейса АОС.....	62
Выводы	62
3 Реализация и тестирование	63
3.1 Структура программных средств	63
3.2 Тестирование программного продукта	65
3.2.1 Тестирование программы формирования расширенной модели	65
3.2.2 Тестирование АОС.....	70
Выводы	79
Заключение	80
Список использованных источников	82
Приложение А. Глоссарий	89
Приложение Б. Исходный код программы.....	91
Приложение В. Связи элементов онтологической модели.....	108
Приложение Г. Скриншоты прототипа аос	115
Приложение Д. Диплом лауреата I степени 2019 г	119
Приложение Е. Диплом лауреата I степени 2020 г.....	121
Приложение Ж. Грамота в номинации «Лучшее оформление»	123
Приложение И. Грамота в номинации «Лучший графический материал»...	125
Приложение К. Свидетельство о регистрации программы	127
Приложение Л. Акт о внедрении.....	129

Введение

Актуальность темы Тенденции развития современного общества, компьютерных информационных технологий и информатизации становятся все более ускоряющимися и с каждым годом охватывают все больше и больше сфер общественной жизни, в том числе и такую важную сферу как образование. В Российской Федерации утвержден такой документ, как Стратегия развития информационного общества в РФ на 2017 – 2030 годы и программа цифровой экономики, в которой отмечены наиболее важные информационные технологии, включая технологии дополненной реальности, а также выделена важная задача применения цифровых технологий в сфере образования в форме автоматизированных обучающих систем (АОС) [1, 2]. Важность эффективного применения новейших информационных технологий в этой сфере, в частности таких как дистанционные образовательные технологии и электронное обучение, отмечается также в Федеральном законе «Об образовании в Российской Федерации» [3].

Таким образом, можно утверждать, что тема диссертации «Исследование и разработка автоматизированных обучающих систем с использованием технологии дополненной реальности» актуальной и перспективной в области информатизации и программной инженерии.

Цель магистерской диссертационной работы состоит в анализе, исследовании, проектировании и последующей реализации АОС с использованием средств онтологического, визуального и математического моделирования на примере АОС по школьному курсу физики.

Задачи исследования. В соответствии с поставленной целью выделены следующие задачи работы:

- а) критический анализ предметной области по тематике диссертации;
- б) актуализация существующей классификации АОС путем добавления новых классификационных признаков и систем;

- в) разработка онтологической модели предметной области;
- г) разработка сетевого графика IT-проекта по созданию программных средств АОС, определение срока и бюджета его выполнения;
- д) разработка модифицированного алгоритма формирования расширенной модели освоения учебного материала для автоматизированной обучающей системы и исследование алгоритма с помощью математического аппарата сетей Петри, а также проведение статистического эксперимента;
- е) регистрация и подготовка к внедрению разработанного и протестированного программного продукта для автоматизированного обучения по школьным курсам.

Объектом исследования является разрабатываемая автоматизированная обучающая система (АОС) по школьному курсу физики.

Предметом исследования являются модели и алгоритмы разработки и применения автоматизированных обучающих систем.

Научная новизна работы заключается в следующем:

- а) предложен модифицированный алгоритм формирования расширенной модели освоения учебного материала, отличающийся от известных тем, что позволяет комплексно учитывать иерархическую структуру учебного материала и эффективно реализовывать варианты траекторий изучения материала при построении гипертекста АОС;
- б) разработана расширенная онтологическая модель предметной области для АОС, отличающаяся от известных более углубленным анализом с целью формализации и систематизации информации для эффективного проведения проектирования объекта исследования;
- в) предложена модифицированная классификация АОС, отличающаяся от известных тем, что предлагается более полный расширенный перечень классификационных признаков и систем.

Методы исследования основаны на современных методах онтологического, визуального моделирования, математических методах сетевого планирования и управления, методах и моделях теории графов, сетей

Петри, методах тестирования и анализа качества программного кода, методах проектирования информационных и программных средств.

Практическая значимость исследований. Разработанная автоматизированная обучающая система протестирована и может быть использована для повышения уровня мультимедийности и интерактивности учебного процесса по школьному курсу физики.

По материалам диссертационной работы получено Свидетельство о государственной регистрации программы для ЭВМ в Федеральном институте промышленной собственности РФ.

Результаты работы внедрены в МБОУ СОШ с углубленным изучением информатики №68 г. Пензы для улучшения структуры материала учебной программы по курсу информатики.

Результаты, выносимые на защиту:

а) предложен и исследован аналитически и экспериментально с помощью математического аппарата сетей Петри модифицированный алгоритм формирования расширенной модели освоения учебного материала, отличающийся от известных тем, что позволяет комплексно учитывать его иерархическую структуру и эффективно реализовывать варианты траекторий изучения материала при построении гипертекста АОС;

б) предложена расширенная онтологическая модель предметной области для АОС, отличающаяся от известных более углубленным анализом с целью формализации и систематизации информации для эффективного проведения проектирования объекта исследования;

в) предложена модифицированная классификация АОС, отличающаяся от известных тем, что предлагается более полный расширенный перечень классификационных признаков и систем;

г) проведен анализ предметной области и требований, проанализированы существующие аналоги, выявлены их преимущества и недостатки, разработаны функциональные и нефункциональные требования к создаваемому программному продукту;

д) было проведено планирование разработки и оценка бюджета программного проекта;

е) было проведено проектирование компонентов программного проекта разработки АОС;

и) были проведены реализация и тестирование компонентов программного проекта разработки АОС.

Апробация работы. Основные результаты работы были доложены и обсуждены на следующих конференциях:

а) Международная научная конференция: Технические и естественные науки. Санкт Петербург, 2020;

б) Международная научная конференция: Наука. Исследования. Практика. Санкт Петербург, 2020;

в) VII Всероссийская межвузовская научно-практическая конференция «Информационные технологии в науке и образовании. Проблемы и перспективы». Пенза, 2020;

г) XXXI Университетская научно-практическая конференция «Актуальные проблемы науки и образования». Пенза, 2021;

д) Международная научно-практическая конференция «Актуальные проблемы науки и образования в условиях современных вызовов». Москва, 2021.

Публикации. По теме диссертационного исследования опубликованы 8 научных работ, из которых одна опубликована в журнале, входящем в перечень ВАК. Получены 2 диплома лауреата 1-й степени и 2 грамоты за победы в номинациях Международных конкурсов научных работ.

Структура и объём работы. Диссертация состоит из введения, трех глав, заключения, списка использованных источников, 10 приложений. Диссертационная работа изложена на 130 страницах, содержит 49 рисунков, 11 таблиц и 42 страницы приложений.

1 Анализ предметной области исследования

1.1 Классификация автоматизированных обучающих систем

1.1.1 Современная классификация АОС

Дадим определение классу АОС или компьютерных средств обучения (далее КСО). К данному классу относятся такие программные средства или программно-технические комплексы, которые предназначены для решения определенных педагогических задач, имеют предметное содержание и ориентированны на взаимодействие с обучаемым [4 – 10].

КСО обладают следующими характерными чертами:

- модульность – возможность формировать и выбирать из набора независимых учебных курсов единый учебный план, отвечающий индивидуальным или групповым потребностям;
- гибкость – возможность заниматься обучением в любом удобном месте в любое время;
- параллельность – возможность обучаться не отрываясь от профессиональной деятельности;
- охват – одновременное обращение ко многим источникам учебной информации большого числа обучающихся;
- экономичность – эффективное использование учебных площадей, технических средств, транспортных средств, концентрированное и унифицированное представление учебной информации и мультидоступ к ней снижает затраты на подготовку специалистов;
- технологичность – внедрение в учебный процесс и использование новейших достижении информационных и телекоммуникационных технологий [8, 9, 10].

В зависимости от решаемых педагогических задач АОС подразделяются на 4 класса (рисунок 1):

- средства теоретической и технологической подготовки;

- средства практической подготовки;
- вспомогательные средства;
- комплексные средства [9, 10].

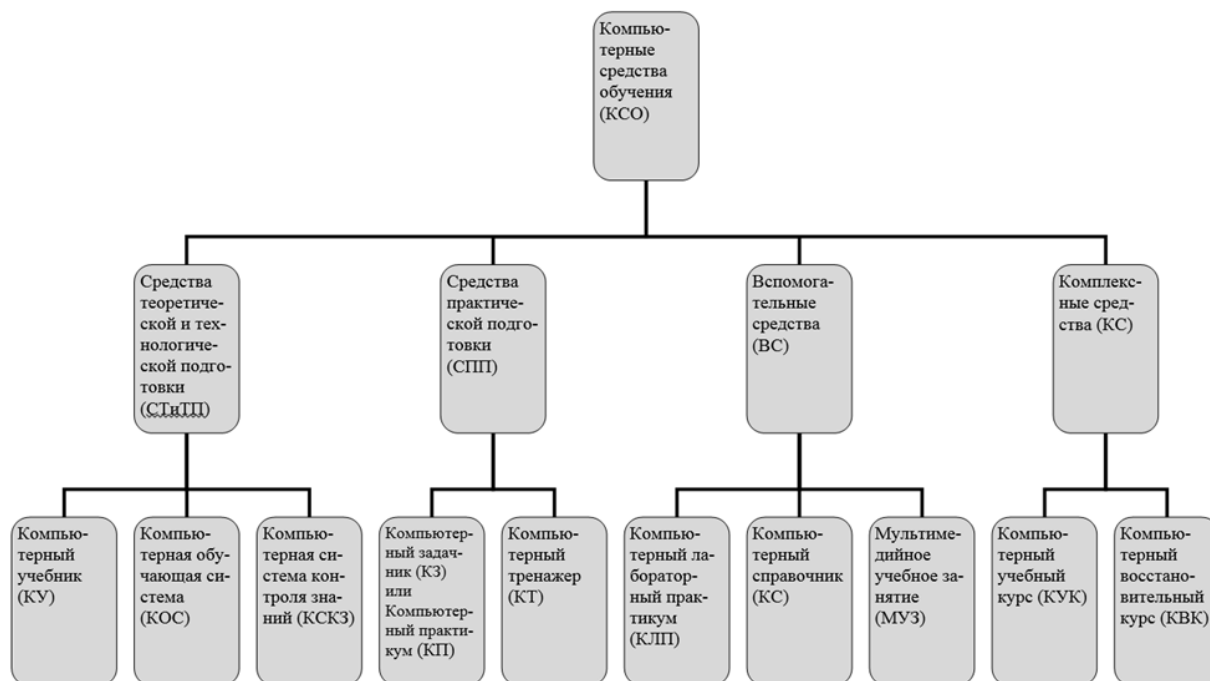


Рисунок 1 – Классификация АОС

К средствам теоретической и технологической подготовки относятся:

- компьютерный учебник – КСО для базовой подготовки по определенному курсу, содержание которого характеризуется относительной полнотой и представлено в форме книги;
- компьютерная обучающая система – КСО для базовой подготовки по одному или нескольким разделам курса;
- компьютерная система контроля знаний – КСО для определения уровня знаний обучаемого по данной дисциплине.

Средства практической подготовки включают два вида систем:

- компьютерный задачник – КСО для выработки умений и навыков решения типовых практических задач в данной ПО;
- компьютерный тренажер – КСО для выработки умений и навыков определенной деятельности, а также развития связанных с ней способностей.

К вспомогательным средствам относят:

- компьютерный лабораторный практикум – КСО для поддержки автоматизированных лабораторных работ;
- компьютерный справочник – КСО, содержащую справочную информационную базу по определенной дисциплине и обеспечивающее возможности ее использования в учебном процессе;
- мультимедийное учебное издание – КСО, основным содержанием которого является мультимедийная запись реального учебного курса или занятия.

В классе комплексных средств, покрывающий широкий круг педагогических задач, особо выделяются два вида КСО:

- компьютерный учебный курс – КСО для подготовки по определенному курсу, в котором интегрированы функции или средства для решения основных задач теоретической, практической и технологической подготовки;
- компьютерный восстановительный курс – КСО для восстановления знаний и умений в рамках определенного курса, в котором интегрированы функции или средства, поддерживающие разные этапы процесса повышения квалификации.

Рассматриваемая выше классификация была представлена еще в 2010 году, поэтому необходимо ее пересмотреть и расширить.

1.1.2 Расширенная классификация АОС

Как уже было сказано в предыдущем разделе, приведенная в предыдущем разделе классификация является не полной и устаревшей. На основании проведенного анализа предметной области, который будет описан далее, предлагается расширенная классификация, которая описывается на описанную выше и представлена на рисунке 2.

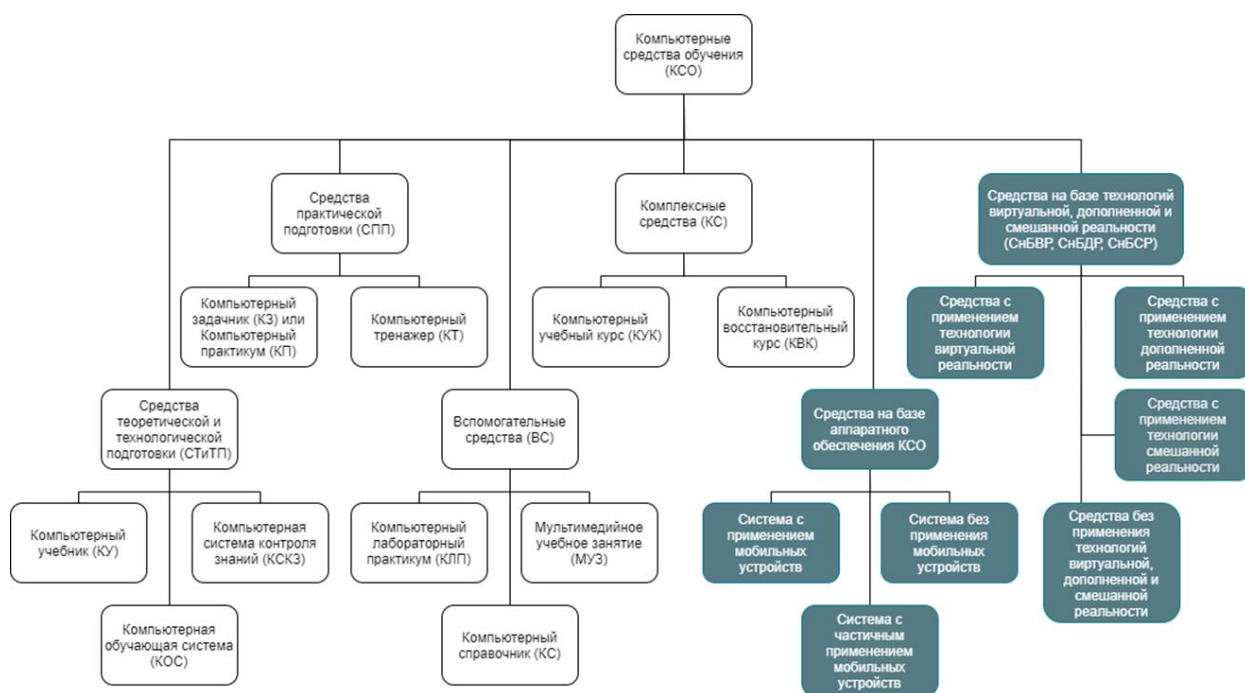


Рисунок 2 – Расширенная классификация АОС

Предлагается расширить приведенную классификацию путем добавления новых классификационных признаков и соответствующих видов АОС. По признаку «средства на базе технологий виртуальной, дополненной и смешанной реальности» предлагается рассматривать следующие виды АОС: АОС с применением технологий виртуальной реальности, АОС с применением технологий дополненной реальности, АОС с применением технологий смешанной реальности, АОС без применения технологий виртуальной, дополненной и смешанной реальности. По признаку «средства на базе аппаратного обеспечения АОС» предлагается рассматривать следующие виды АОС: АОС с применением мобильных устройств, АОС без применения мобильных устройств, АОС с частичным применением мобильных устройств.

Предложенная расширенная классификация АОС позволяет учитывать современные тенденции, а области информационных систем и технологий и программной инженерии.

1.2 Анализ аналогичных автоматизированных обучающих систем и/или лабораторных практикумов по физике

Прежде чем начать разрабатывать свой продукт, всегда сначала необходимо изучить его аналоги и выделить их преимущества, с целью их дальнейшего применения, и недостатки, для их исправления. Для разрабатываемой системы автоматизированного обучения такими аналогами являются компьютерный лабораторный практикум по физике института дистанционного образования [11], справочники и лабораторные работы с сайта all-fizika.com [12], мобильное приложение «Физика от А до Я, помощник ЕГЭ» [13].

1.2.1 Компьютерный лабораторный практикум по физике института дистанционного образования

Данный проект представляет собой web-приложение – справочник, хранящий в себе методические указания по лабораторным работам из всех разделов физики. Его интерфейс представлен на рисунке 3. К каждому из этих указаний привязана ссылка для скачивания написанных на Delphi оконных приложений, имитирующих выполнение каждой из лабораторных работ. Интерфейс одного из этих приложений представлен на рисунке 4.

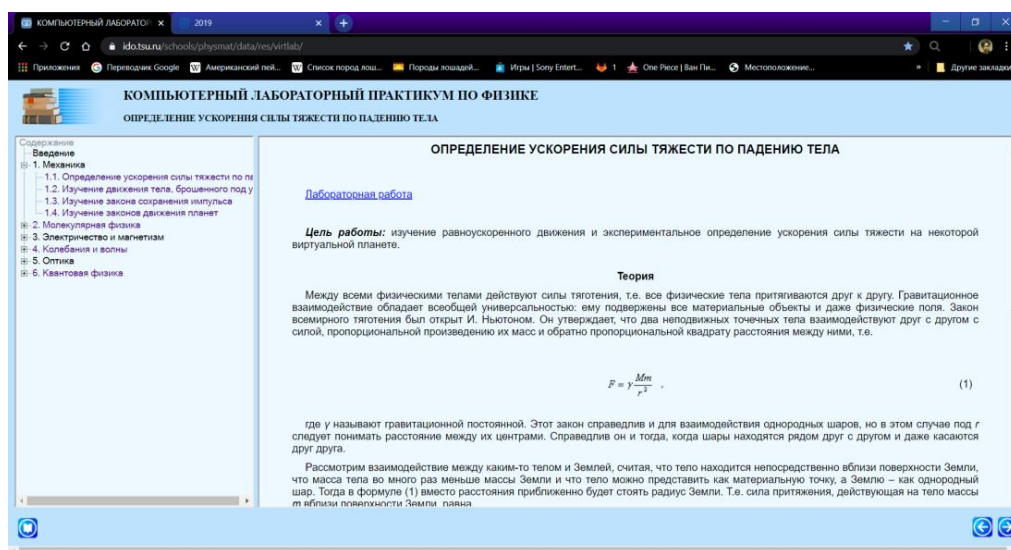


Рисунок 3 – Интерфейс web-справочника

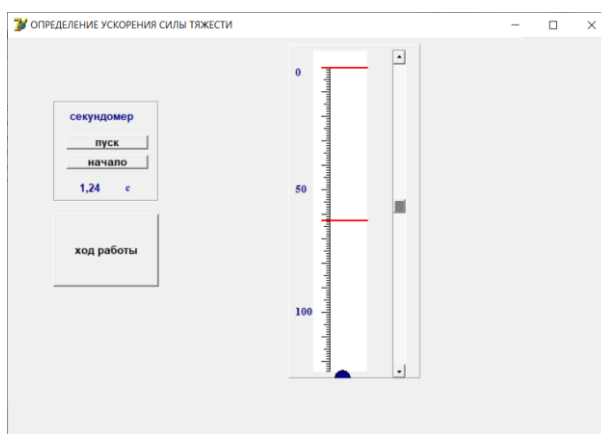


Рисунок 4 – Интерфейс приложения для выполнения лабораторных работ

К плюсам данного продукта можно отнести удобное боковое меню, реализованное в виде раскрывающегося древовидного списка. Все темы в нем удобно отсортированы по разделам.

Минусами же данного приложения являются отсутствие писания онтологического моделирования, неэффективная организация структуры самого учебного материала внутри каждой темы и отсутствие гиперссылок, что затрудняет повторение или повторное изучение позабытых тем.

Так же весомым минусом являются интуитивно непонятный и неудобный двухмерный интерфейс программ для выполнения лабораторного практикума, который не передает представления об использовании приборов или проведения подобных опытов в реальной жизни.

1.2.2 Компьютерный справочник и лабораторные работы с сайта all-fizika.com

Еще одним похожим приложением является онлайн справочник по физике, размещенный на сайте all-fizika.com. Он содержит справочную информацию, определения, статьи и лекции по всем разделам физики. Вся справочная информация на этом сайте разбита на несколько разных категорий, поиск внутри которых организован разными способами. Так, в разделе «Энциклопедия», с помощью которого можно найти определения для каждого из физических терминов, поиск осуществляется с помощью алфавитного

указателя. А в разделе «Лекции» все материалы структурированы по темам и располагаются в наилучшем для изучения порядке. Интерфейс сайта представлен на рисунке 5.

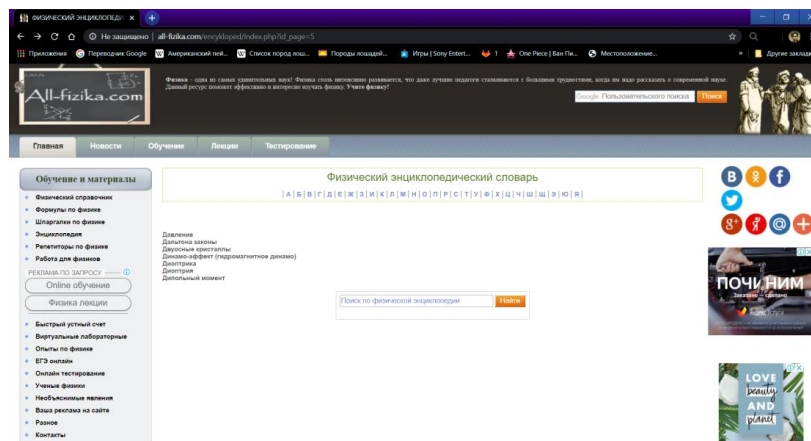


Рисунок 5 – Интерфейс сайта all-fizika.com

Помимо теоретической информации, на сайте есть раздел с Flash-приложениями, имитирующими выполнения различных лабораторных работ. Интерфейс одного из них представлен на рисунке 6.

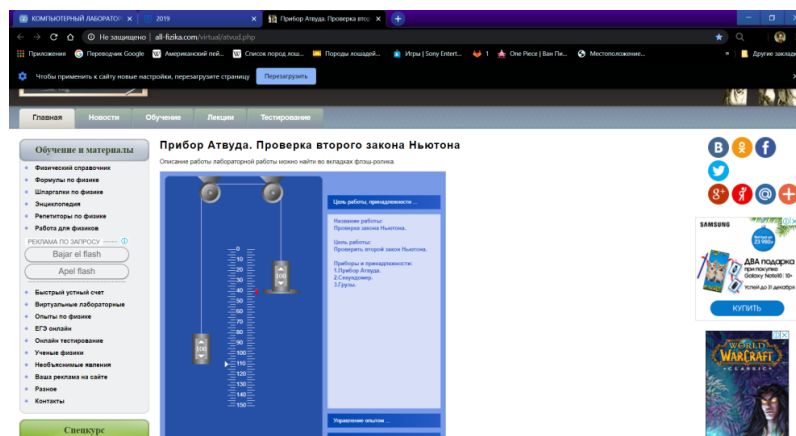


Рисунок 6 – Интерфейс приложения для выполнения лабораторных работ

Плюсами данного приложения можно отнести большой объем грамотно структурированной информации.

Среди минусов можно назвать все то же отсутствие гиперссылок внутри статей, что, как и в предыдущем случае, замедляет изучение и освоение материала.

Несмотря на то, что графически интерфейс приложений для выполнения лабораторных работ у данного продукта более красочный и удобный, чем в предыдущем случае, он все так же слишком схематично описывает процесс их выполнения и не дает пользователям реалистичного опыта.

Так же минусом является то, что у лабораторных работ отсутствуют ссылки на материал соответствующих тем, что усложнит их выполнение.

1.2.3 Справочник по физике для смартфонов «Физика от А до Я, помощник ЕГЭ»

Из аналогов справочной системы на целевой платформе – смартфонах под управлением операционной системы Android можно выделить справочник «Физика ЕГЭ». Он имеет удобный для работы на смартфоне интерфейс, упорядоченность материалов по разделам. На рисунке 7 представлен интерфейс этого приложения.

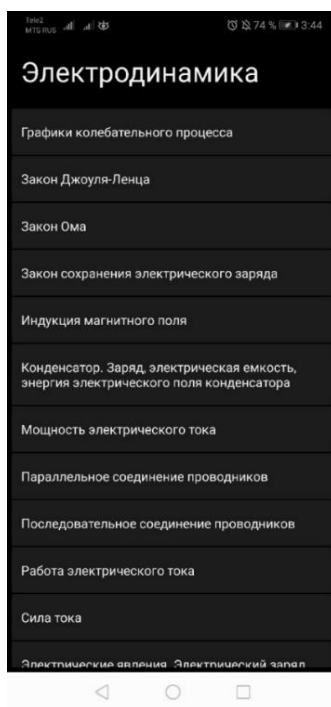


Рисунок 7 – Интерфейс приложения «Физика ЕГЭ»

Основными минусами этого приложения являются слишком краткое и скудное изложение информации и, также, как и в предыдущем примере,

отсутствие гиперссылок внутри каждой статьи. Важным минусом является и тот факт, что справочник не связан с конкретными лабораторными работами и предоставляет лишь общую информацию.

1.2.4 Общие выводы по анализу аналогов

Все перечисленные выше аналоги имеют ряд общих недостатков. У всех не очень грамотно организованы гиперссылки внутри текста, что затрудняет изучение материала. Ни одна из рассматриваемых АОС не имеет сведений о проведении онтологического моделирования.

У тех из них, которые имеют лабораторный практикум, его реализация так же имеет существенные недостатки. Самый важный из этих недостатков – схематичность изображения процесса выполнения лабораторных работ, которая не позволяет пользователем получить представление об использовании приборов или проведения подобных опытов в реальной жизни. Причиной этого является использование двухмерного интерфейса. Эту проблему поможет решить переход к использованию трехмерного интерфейса с использованием технологии дополненной реальности [14 – 24].

1.3 Модель освоения учебного материала

1.3.1 Общее описание модели освоения учебного материала

Для устранения главного недостатка всех перечисленных выше решений – отсутствия гиперссылок внутри текста, а также для лучшей структуризации материала с точки зрения последовательности его изучения возьмем за основу модель освоения учебного материала, описанной в работе [25].

В состав модели освоения входят матрицы уровней, отношений очередности и логических связей учебных элементов, а также строящиеся на

их основе графы отношений очередности и логических связей учебных элементов. Построение модели производится в четыре этапа:

- а) формирование матрицы отношений очередности учебных элементов с учетом уровней учебных элементов;
- б) обработка матрицы отношений очередности и построение последовательности изучения учебного материала в виде списка учебных элементов;
- в) формирование матрицы логических связей учебных элементов;
- г) построение графа логических связей учебных элементов.

Первый и третий этапы являются неформальными и выполняются на основе анализа учебного материала. Матрицы отношений очередности A и логических связей учебных элементов B являются квадратными. Размер матриц равен количеству учебных элементов n :

$$A[i, j]: i = 1..n, j = 1..n; B[i, j]: i = 1..n, j = 1..n.$$

Сначала строят ячейки матриц и нумеруют их строки и столбцы в соответствии с возрастанием учебных элементов. Далее построчно заполняют ячейки матриц нулями и единицами.

При заполнении ячеек матрицы отношений очередности анализируют простое бинарное отношение очередности между двумя учебными элементами. Единицу ставят в ячейку, если учебный элемент, указанный в номере строки, должен изучаться после учебного элемента, указанного в номере столбца. Противоположное отношение очередности обозначают нулем:

$$A[i, j] = \begin{cases} 1, & \text{если элемент } i \text{ изучается после } j, \\ 0 & \text{в противном случае} \end{cases}.$$

Все ячейки главной диагонали матрицы отношений очередности заполняют единицами. Ячейки матрицы, симметричные относительно главной диагонали, должны иметь противоположные отношения (0 или 1). Поэтому неформальный анализ парных отношений очередности можно проводить

лишь для левого нижнего или для правого верхнего треугольника матрицы, заполняя ее оставшуюся часть формально на основе свойства антисимметрии.

При заполнении матрицы логических связей учебных элементов ставят единицу в ячейку, если учебный материал элемента, указанного в номере строки, логически связан с учебным материалом элемента, указанного в номере столбца:

$$B[i, j] = \begin{cases} 1, & \text{если элемент } i \text{ логически связан с } j, \\ 0 & \text{в противном случае} \end{cases}.$$

Составление матрицы логических связей удобно вести на основе матрицы отношений очередности путем исключения единиц из тех ячеек, для которых отсутствуют логические, опорные связи между элементами.

Процесс заполнения матриц целесообразно вести на основе таблицы учебных элементов и текстов с учебным материалом по всем учебным элементам. Анализ содержания учебного материала позволяет более объективно выявлять парные отношения очередности и логические связи между учебными элементами.

На вид матриц отношений очередности и логических связей, а, следовательно, в дальнейшем и на форму представления учебного материала оказывают влияние не только объективные, но и субъективные факторы: вкусы разработчика обучающей системы, его привычки, интуитивные представления, склад мышления и т.д.

Последовательность изучения учебного материала в пошаговой процедуре обучения определяется в процессе формальной обработки матрицы отношений очередности путем суммирования элементов каждой строки матрицы:

$$S_i = \sum_{j=1}^n A[i, j].$$

Полученные суммы записывают в колонке справа от матрицы. Величины сумм указывают порядковые номера соответствующих учебных элементов в списке последовательности изучения учебного материала.

Логические связи учебных элементов отображают для наглядности в виде ориентированного графа

$$G(V, E) = \langle V, E \rangle, V \neq \emptyset, E \subset V \times V.$$

Строят граф по матрице логических связей учебных элементов, которая является для него транспонированной матрицей смежности. Целесообразно располагать этот граф под списком последовательности учебных элементов, сохраняя указанный в списке порядок освоения учебного материала. Дуги графа логических связей указывают на опорные связи между учебными элементами.

Описанная модель не является идеальной и имеет свои недостатки. Она применяется отдельно для учебных элементов только одного уровня, и при сложной иерархической структуре учебного материала это вызывает определенные трудности ее применения. К тому же указанная модель должна быть согласована с другими моделями, в частности с моделью содержания учебного материала, что также приводит к дополнительным трудностям ее использования.

1.3.2 Расширенная модель освоения учебного материала

Для решения описанных в предыдущем разделе недостатков модели предлагается расширенная модель освоения учебного материала, лишенная перечисленных недостатков.

В качестве иллюстрации использования расширенной модели освоения учебного материала приведем ее применение по отношению к учебному материалу по разделу школьного курса физики «Механика».

Данный материал имеет следующую иерархическую структуру.

1 – статика (определение);

1.1 – равновесие тел (определение);

- 1.1.1 – момент силы;
- 1.1.2 – условие равновесия тел;
- 1.1.3 – виды равновесия;
- 1.2 – гидростатика и аэродинамика (определение);
- 1.2.1 – основные понятия;
- 1.2.1.1 – давление жидкости и газов;
- 1.2.1.2 – закон Паскаля;
- 1.2.1.3 – сообщающиеся сосуды;
- 1.2.1.4 – уравнение Бернулли;
- 1.2.2 – закон Архимеда;
- 1.2.2.1 – условия плавания тел;
- 1.2.3 – гидравлические машины;
- 2 – колебания и волны (основные понятия);
- 2.1 – механические колебания;
- 2.2 – гармонические колебания;
- 2.3 – резонанс;
- 2.4 – автоколебания;
- 2.5 – механические волны;
- 2.6 – звуковые волны.

Далее были выделены и пронумерованы следующие учебные элементы:

- 1 – статика (определение);
- 2 – равновесие тел (определение);
- 3 – момент силы;
- 4 – условие равновесия тел;
- 5 – виды равновесия;
- 6 – гидростатика и аэродинамика (определение);
- 7 – основные понятия;
- 8 – давление жидкости и газов;
- 9 – закон Паскаля;
- 10 – сообщающиеся сосуды;

- 11 – уравнение Бернулли;
- 12 – закон Архимеда;
- 13 – условия плавания тел;
- 14 – гидравлические машины;
- 15 – колебания и волны (основные понятия);
- 16 – механические колебания;
- 17 – гармонические колебания;
- 18 – резонанс;
- 19 – автоколебания;
- 20 – механические волны;
- 21 – звуковые волны.

После этого по выделенным учебным элементам был построен граф отношений очередности учебных элементов с учетом их нумерации, представленный на рисунке 8. Также была сформирована таблица уровней учебных элементов, представленная в таблице 1.

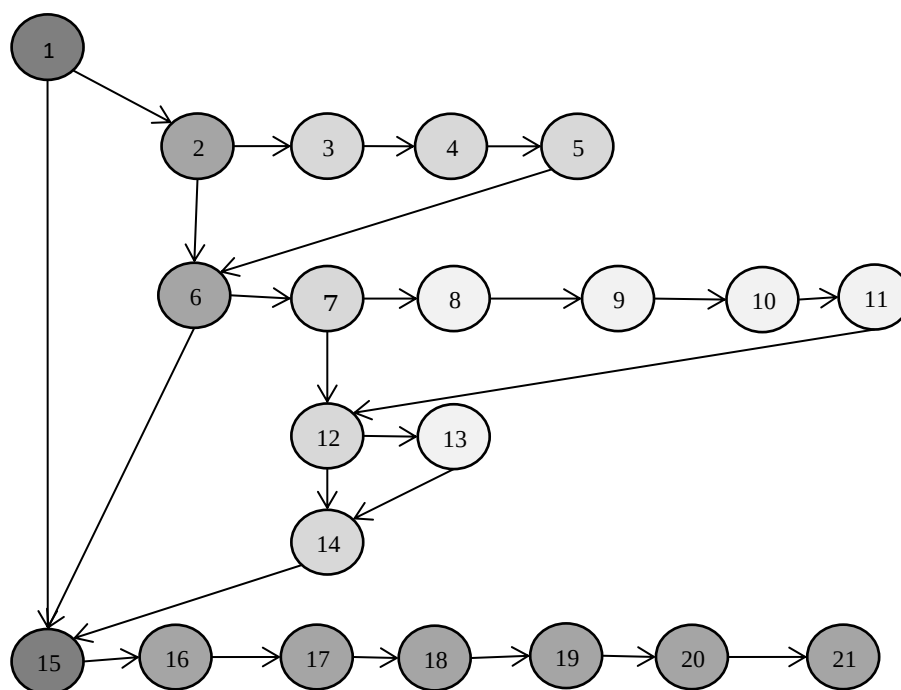


Рисунок 8 – Граф отношений очередности учебных элементов

Таблица 1 – Таблица уровней учебных элементов

Номер вершины в графе i	Уровень	Номер вершины, определяющей уровень
1	1	-
2	2	1
3	3	2
4	3	2
5	3	2
6	2	1
7	3	6
8	4	7
9	4	7
10	4	7
11	4	7
12	3	6
13	4	12
14	3	6
15	1	-
16	2	15
17	2	15
18	2	15
19	2	15
20	2	15
21	2	15

Затем на основании полученных данных была построена матрица отношений очередности учебных элементов, изображенная на рисунке 9.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Сум ма
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
4	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
5	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
6	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
7	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
8	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	8
9	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	9
10	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	10
11	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	11
12	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	12
13	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	13
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	14
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	15
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	16
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	17
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	18
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	19
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	20
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	21

Рисунок 9 – Таблица отношений очередности учебных элементов

Далее были сформированы таблица отношений логических связей учебных элементов, представленная на рисунке 10, и общий граф логических связей учебных элементов, отображенный на рисунке 11, который определяет последовательность изложения материала, варианты траекторий изучения, логические связи.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Рисунок 10 – Таблица отношений логических связей учебных элементов

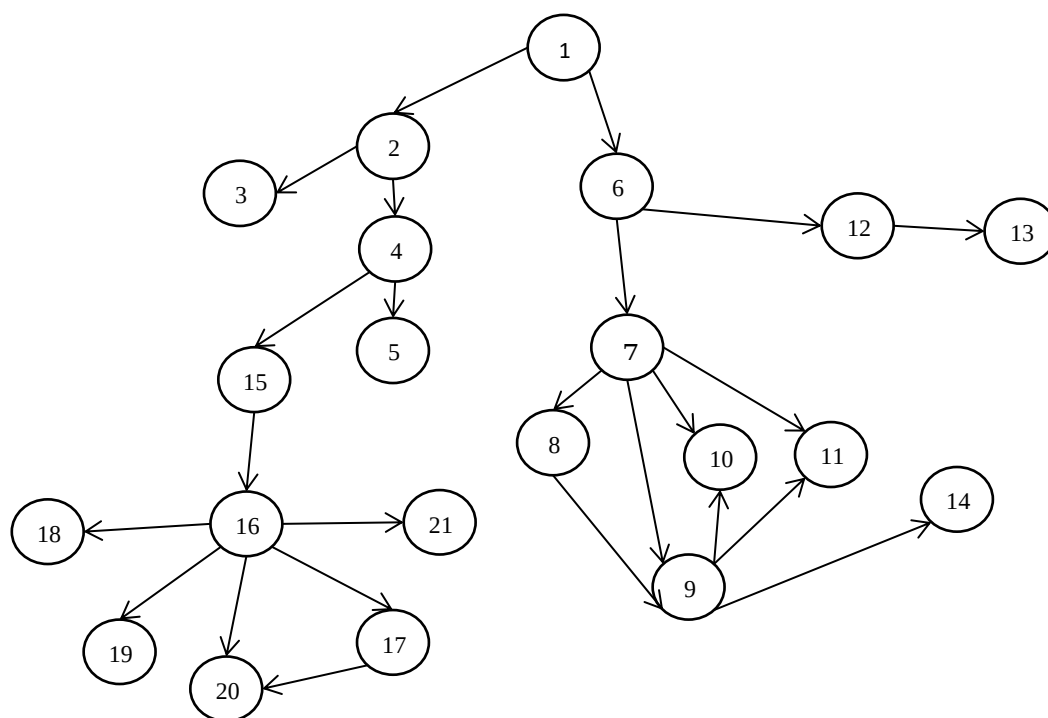


Рисунок 11 – Общий граф отношений логических связей учебных элементов

Таким образом, предложена расширенная модель освоения учебного материала, позволяющая комплексно учитывать его иерархическую структуру и эффективно реализовывать варианты траекторий изучения материала при построении гипертекста АОС [26 – 32].

1.4 Планирование разработки и технико-экономическое обоснование

Всегда для того, чтобы доказать экономическую целесообразность проекта, прежде чем приступить к его выполнению, нужно оценить его трудоемкость и себестоимость. Для этого был определен список задач, решение которых необходимо для создания приложения, а также был выделен перечень ресурсов, с помощью которых эти задачи будут решаться.

В таблице 2 представлены задачи, которые были определены в качестве этапов создания проекта во время выполнения этапа его технико-экономического обоснования.

Таблица 2 – Перечень задач проекта

№	Название задачи	Длительность	Предшественники
1	Начало проекта	–	–
2	Разработка расширенной модели освоения УМ	–	–
3	Поиск литературы по моделям усвоения УМ	3	1
4	Консультация с преподавателями	1	3
5	Анализ и выявление недочетов найденных моделей	4	4
6	Выбор расширяемой модели	1	5
7	Разработка способов расширения модели	7	6
8	Выбор лучшего способа расширения модели	1	7
9	Разработка расширенной модели УМ завершена	–	8

Продолжение таблицы 2

№	Название задачи	Длительность	Предшественники
10	Анализ требований	–	–
11	Поиск литературы (учебников физики и учебных пособий)	7 дней	1
12	Составление списка лабораторных работ и задействованных в них приборов	4 дня	11
13	Анализ списка лабораторных работ и приборов	2 дня	12
14	Консультация с учителями физики	1 день	13
15	Принятие окончательного решения	1 день	14
16	Анализ требований завершен	–	15
17	Выбор средств реализации проекта	–	–
18	Анализ возможных программных средств	5 дней	9;16
19	Выбор программного средства для реализации	2 дня	18
20	Выбор средств реализации завершен	–	19
21	Реализация проекта	–	–
22	Проектирование пользовательского интерфейса	4 дня	16;20
23	Создание и анимирование 3D моделей	9 дней	16;20
24	Интеграция 3D моделей в среду	5 дней	23
25	Программирование физики поведения моделей	–	–
26	Создание математических моделей приборов	7 дней	24
27	Реализация математических моделей	12 дней	26
28	Программирование сопутствующих и обслуживающих классов	19 дней	22
29	Реализация проекта завершена	–	27;28
30	Отладка	–	–

Окончание таблицы 2

31	Тестирование работы отдельных приборов	5 дней	29
32	Тестирование работы сопутствующих классов	5 дней	29
33	Исправление ошибок	7 дней	31;32
34	Отладка завершена	–	33
35	Внедрение программного средства в общий проект	–	–
36	Внедрение программного средства	3 дня	34
37	Поиск и исправление ошибок при внедрении	4 дня	36
38	Внедрение завершено	–	37
39	Конец проекта	–	38

Из этой таблицы можно сделать вывод, что для реализации программного проекта нужно пройти через решение 39 задач.

В таблице 3 представлен список затрачиваемых ресурсов.

Таблица 3 – Список ресурсов проекта

Название ресурса	Тип	Стандартная ставка	Ставка сверхурочных
Программист1	Трудовой	20 Р\ч	40 Р\час
Программист2	Трудовой	20 Р\ч	40 Р\час
Интернет	Затраты		
Компьютер	Материальны	200 Р	
Android-устройство	Материальный	200 Р	

Из представленных таблиц можно увидеть, что решением задач будут заниматься 3 программиста, которым для выполнения поставленных задач необходимы компьютеры, постоянных доступ к сети интернету, а также

возможность использовать Android-устройство для тестирования приложения по ходу его разработки.

Сначала вручную было просчитано время, которое необходимо для реализации проекта и определен его критический путь для определения границ сроков.

Через t_i обозначено время выполнения работы i , G – множество работ, непосредственно предшествующих работе i , H – множество работ, непосредственно следующих за работой i .

Для вычисления критического пути необходимо для каждой задачи определить раннее и позднее время начала работ. Раннее время определяется по формуле:

$$T_{p(i)} = \max_{j \in G} (T_{p(j)} + t_j).$$

Позднее время начала следует вычислять по формуле:

$$T_{п(i)} = \min_{j \in H} (T_{п(j)} - t_j).$$

Вычисление раннего времени начала работ расписано в таблице 4.

Таблица 4 – Вычисление раннего времени начала работ

Работа	Предшественники			Вычисление	Раннее время начала
	№	Раннее время начала	Длительность		
1	–	–	–	–	0
3	1	0	0	0+0	0
4	3	0	3	0+3	3
5	4	3	1	3+1	4
6	5	4	4	4+4	8
7	6	8	1	8+1	9
8	7	9	7	9+7	16
9	8	16	1	16+1	17
11	1	0	0	0+0	0
12	11	0	7	0+7	7

Продолжение таблицы 4

Работа	Предшественники			Вычисление	Раннее время начала
	№	Раннее время начала	Длительность		
13	12	7	4	7+4	11
14	13	11	2	11+2	13
15	14	13	1	13+1	14
16	15	14	1	14+1	15
18	9	17	0	Max(17+0;15+0)	17
	16	15	0		
19	18	17	5	17+5	22
20	19	22	2	22+2	24
22	16	15	0	Max(15+0;24+0)	24
	20	24	0		
23	16	15	0	Max(15+0;24+0)	24
	20	24	0		
24	23	24	9	24+9	33
26	24	33	5	33+5	38
27	26	38	7	38+7	45
28	22	24	4	24+4	28
29	27	45	12	Max(45+12;28+19)	57
	28	28	19		
31	29	57	0	57+0	57
32	29	57	0	57+0	57
33	31	57	5	Max(57+5;57+5)	62
	32	57	5		
34	33	62	7	62+7	69
36	34	69	0	69+0	69
37	36	69	3	69+3	72
38	37	72	4	72+4	76
39	38	16	0	76+0	76

Из этих вычислений можно увидеть, что выполнение проекта должно занять 76 рабочих дней. Теперь, для дальнейшего определения критического пути, необходимо вычислить время позднего начала работ. Оно представлено в таблице 5.

Таблица 5 – Вычисление позднего времени начала работ

Работа	Последователи		Длительность	Вычисление	Позднее время начала
	№	Позднее время начала			
39	-	-	-	-	76
38	39	76	0	76-0	76
37	38	76	4	76-4	72
36	37	72	3	72-3	69
34	36	69	0	69-0	69
33	34	69	7	69-7	62
32	33	62	5	62-5	57
31	33	62	5	62-5	57
29	31 32	57 57	0	Min(57-0;57-0)	57
28	29	57	19	57-19	38
27	29	57	12	57-12	45
26	27	45	7	45-7	38
24	26	38	5	38-5	33
23	24	33	9	33-9	24
22	28	38	4	38-4	34
20	22 23	34 24	0	Min(34-0;24-0)	24
19	20	24	2	24-2	22
18	19	22	5	22-5	17
16	18	17	0	17-0	17
15	16	17	1	17-1	16
14	15	16	1	16-1	15

Продолжение таблицы 5

Работа	Последователи		Длительность	Вычисление	Позднее время начала
	№	Позднее время начала			
13	14	15	2	15-2	13
12	13	13	4	13-4	9
11	13	9	7	9-7	2
9	18	17	0	17-0	17
8	9	17	1	17-1	16
7	8	16	7	16-7	9
6	7	9	1	9-1	8
5	6	8	4	8-4	4
4	5	4	1	4-1	3
3	4	3	3	3-3	0
1	3	0	0	Min(0-0;2-0)	0
	11	0			

Зная оба вида начала работ, можно определить критический путь для разрабатываемого проекта. Все необходимые для этого вычисления представлены в таблице 6.

Таблица 6 – Вычисление критического пути

Работа №	Раннее время начала	Позднее время начала	Резерв времени
1	0	0	0
3	0	0	0
4	3	3	0
5	4	4	0
6	8	8	0
7	9	9	0
8	16	16	0
9	17	17	0
11	0	2	2

Продолжение таблицы 6

Работа №	Раннее время начала	Позднее время начала	Резерв времени
12	7	9	2
13	11	15	4
14	13	15	2
15	14	16	2
16	15	17	2
18	17	17	0
19	22	22	0
20	24	24	0
22	24	34	10
23	24	24	0
24	33	33	0
26	38	38	0
27	45	45	0
28	28	38	10
29	57	57	0
31	57	57	0
32	57	57	0
33	62	62	0
34	69	69	0
36	69	69	0
37	72	72	0
38	76	76	0
39	76	76	0

Из этой таблицы можно определить, что критический путь проходит через работы, с номерами 1, 3, 4, 5, 6, 7, 8, 9, 18, 19, 20, 23, 24, 26, 27, 29, 31, 32, 33, 34, 36, 37, 38, 39.

На рисунке 12 изображено введенное в Project содержание таблицы 3 с вычисленными датами начала и завершения каждой задачи. За начало проекта была принята дата 27.08.2020.

Режим задачи	Название задачи	Длительность	Начало	Окончание	Предшественники	Названия ресурсов	Свойства новой строки
✓	Начало проекта	0 дней	Чт 27.08.20	Чт 27.08.20			
✓	Разработка расширенной модели освоения УМ	17 дней	Пн 28.08.20	Вт 22.09.20			
✓	Поиск литературы по модели освоения УМ	3 дней	Пн 28.08.20	Вт 01.09.20	1	Программист1,Компьютер[1]	
✓	Консультация с преподавателями	1 день	Ср 02.09.20	Ср 02.09.20	3	Программист2,Компьютер[1]	
✓	Анализ и выявление недостатков найденных моделей	4 дней	Чт 03.09.20	Вт 08.09.20	4	Программист1,Компьютер[1]	
✓	Выбор расширенной модели	1 день	Ср 09.09.20	Ср 09.09.20	5	Программист1,Компьютер[1]	
✓	Разработка способов расширения модели	7 дней	Чт 10.09.20	Пн 18.09.20	6	Программист1,Компьютер[1]	
✓	Выбор лучшего способа расширения модели	1 день	Пн 21.09.20	Пн 21.09.20	7	Программист1,Компьютер[1]	
✓	Разработка расширенной модели УМ завершена	0 дней	Вт 22.09.20	Вт 22.09.20	8		
✓	Анализ требований	15 дней	Пн 28.08.20	Пн 18.09.20			
✓	Поиск литературы (учебников физики и учебных пособий)	7 дней	Пн 28.08.20	Пн 07.09.20	1	Программист2,Компьютер[1]	
✓	Составление списка лабораторных работ и задействованных в них приборов	4 дней	Вт 08.09.20	Пн 11.09.20	11	Программист2,Компьютер[1]	
✓	Анализ списка лабораторных работ и приборов	2 дней	Пн 14.09.20	Вт 15.09.20	12	Программист1,Компьютер[1]	
✓	Консультация с учителями физики	1 день	Ср 16.09.20	Ср 16.09.20	13	Программист2,Компьютер[1]	
✓	Принятие окончательного решения	1 день	Чт 17.09.20	Чт 17.09.20	14	Программист2,Компьютер[1]	
✓	Анализ требований завершён	0 дней	Пн 18.09.20	Пн 18.09.20	15		
✓	Выбор средств реализации проекта	7 дней	Вт 22.09.20	Чт 01.10.20			
✓	Анализ возможных программных средств	5 дней	Вт 22.09.20	Пн 28.09.20	9,16	Программист1[50%],Програ	
✓	Выбор программного средства для реализации	2 дней	Вт 29.09.20	Ср 30.09.20	18	Программист1[50%],Програ	
✓	Выбор средств реализации завершён	0 дней	Чт 01.10.20	Чт 01.10.20	19		
✓	Реализация проекта	33 дней	Пн 02.10.20	Ср 18.11.20			
✓	Проектирование пользовательского интерфейса	4 дней	Пн 02.10.20	Ср 07.10.20	16,20	Программист2,Компьютер[1]	
✓	Создание и анимирование 3D моделей	9 дней	Пн 02.10.20	Ср 14.10.20	16,20	Программист1,Компьютер[1]	
✓	Интеграция 3D моделей в среду	5 дней	Чт 15.10.20	Ср 21.10.20	23	Программист2,Компьютер[1]	
✓	Программирование физики поведения моделей	19 дней	Чт 22.10.20	Вт 17.11.20			
✓	Программирование сопутствующих и обучающих классов	19 дней	Чт 08.10.20	Вт 03.11.20	22	Программист2,Компьютер[1]	
✓	Реализация проекта завершена	0 дней	Ср 18.11.20	Ср 18.11.20	27,28		
✓	Отладка	12 дней	Чт 19.11.20	Пн 07.12.20			
✓	Тестирование работы отладочных приборов	5 дней	Чт 19.11.20	Ср 25.11.20	29	Программист1,Компьютер[1]	
✓	Тестирование работы сопутствующих классов	5 дней	Чт 19.11.20	Ср 25.11.20	29	Программист2,Компьютер[1]	
✓	Исправление ошибок	7 дней	Чт 26.11.20	Пн 04.12.20	31,32	Программист1[50%],Програ	
✓	Отладка завершена	0 дней	Пн 07.12.20	Пн 07.12.20	33		
✓	Внесение программного средства в общий проект	7 дней	Вт 08.12.20	Чт 17.12.20			
✓	Выдача программного средства	3 дня	Вт 08.12.20	Чт 10.12.20	34	Программист1[50%],Програ	
✓	Поиск и исправление ошибок при вводе	4 дня	Пн 11.12.20	Ср 16.12.20	36	Программист1[50%],Програ	
✓	Внесение завершено	0 дней	Чт 17.12.20	Чт 17.12.20	37		
✓	Конец проекта	0 дней	Пн 18.12.20	Пн 18.12.20	38		

Рисунок 12 – Задачи проекта, введенные в среду Microsoft Project 2010

На рисунке 13 представлена построенная программой диаграмма Ганта.

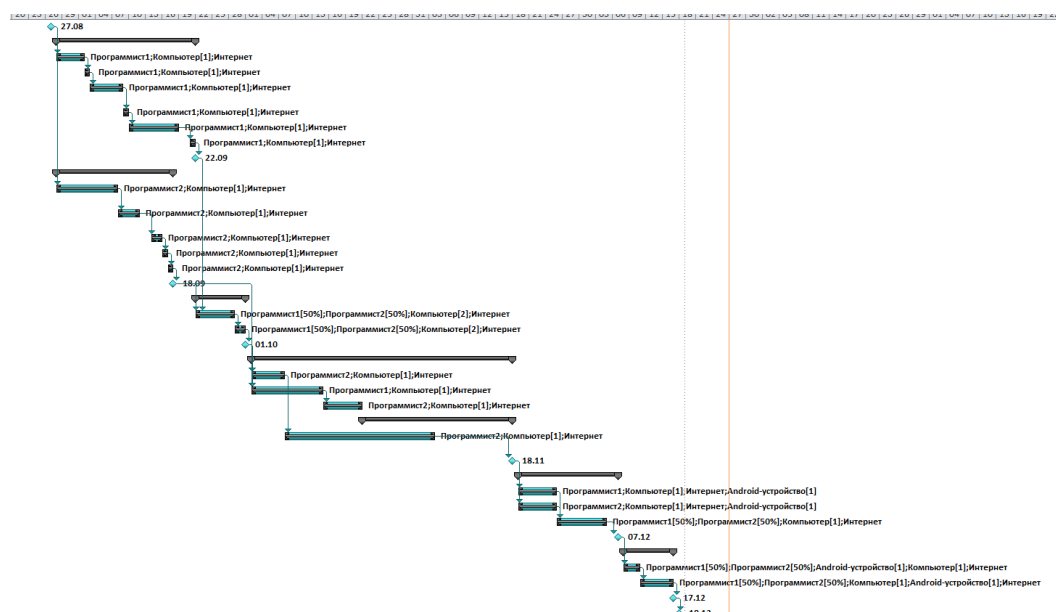


Рисунок 13 – Диаграмма Ганта

Из этих рисунков можно понять, что вычисленный программой критический путь проходит по задачам 1, 3, 4, 5, 6, 7, 8, 9, 18, 19, 20, 23, 24, 26, 27, 29, 31, 32, 33, 34, 36, 37, 38, 39.

На рисунке 14 изображен обзор проекта, на котором можно увидеть подсчитанную Microsoft Project 2010 длительность реализации проекта – 81 день и предполагаемые затраты – 25040 руб. Из этого можно сделать вывод об экономической целесообразности проекта.

<u>Даты</u>			
Начало:	Чт 27.08.20	Окончание:	Пт 18.12.20
Баз. начало:	НД	Баз. окончание:	НД
Факт. начало:	НД	Факт. окончание:	НД
Отклон. начало:	0 дней	Отклон. окончание:	0 дней
<u>Длительность</u>			
Текущий план:	81 дней	Оставшаяся:	81 дней
Базовый план:	0 дней	Фактическая:	0 дней
Отклонение:	81 дней	Процент завершения:	0%
<u>Трудозатраты</u>			
Текущий план:	952 ч	Оставшиеся:	952 ч
Базовый план:	0 ч	Фактические:	0 ч
Отклонение:	952 ч	Процент завершения:	0%
<u>Затраты</u>			
Текущий план:	25 040,00 Р	Оставшиеся:	25 040,00 Р
Базовый план:	0,00 Р	Фактические:	0,00 Р
Отклонение:	25 040,00 Р		
<u>Состояние задачи</u>		<u>Состояние ресурса</u>	
Еще не начатые задачи:	39	Трудовые ресурсы:	2
Выполняющиеся задачи:	0	Трудовые ресурсы с преем. до:	1
Завершенные задачи:	0	Материальные ресурсы:	2
Итого задач:	39	Итого ресурсов:	5

Рисунок 14 – Обзор проекта

Ниже приведены графики ресурсов. На рисунке 15 представлен график для материального ресурса «Компьютер».

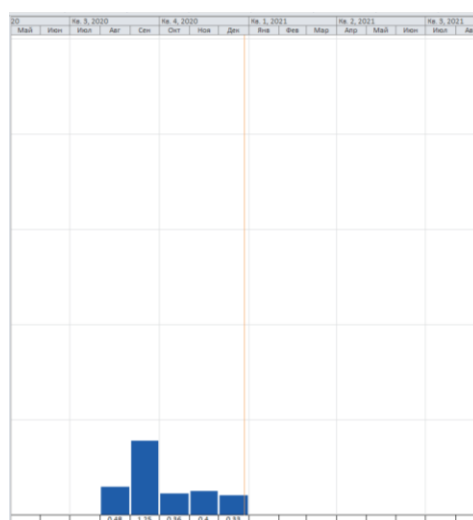


Рисунок 15 – График загрузки для ресурса «Компьютер»

На рисунках 16 и 17 представлены графики для трудовых ресурсов «Программист1» и «Программист2».

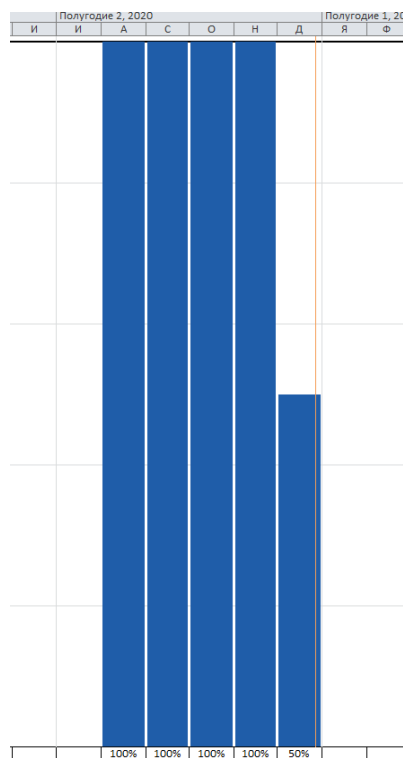


Рисунок 16 – График загрузки для ресурса «Программист1»

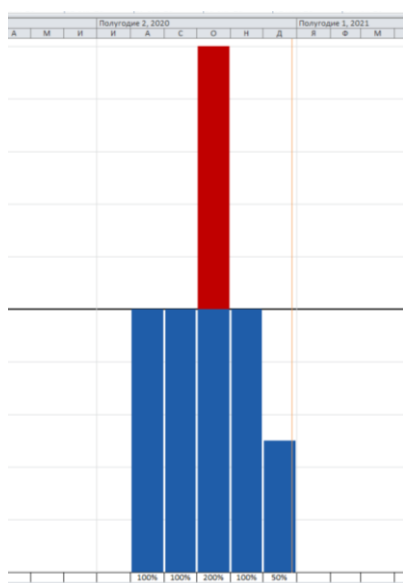


Рисунок 17 – График загрузки для ресурса «Программист2»

Из этих рисунков видно, для ресурса «Программист2» в октябре нагрузка превышает нормальную.

1.5 Выбор технологии разработки

В качестве технологии разработки была выбрана технология RUP – Rational Unified Process. В ее основе лежит итеративный подход к проектированию и разработке программных средств, основанный на спиральном жизненном цикле.

В технологии RUP большое внимание уделяется начальным этапам разработки (анализ требований и моделирование), что позволяет уже на ранних стадиях обнаруживать ошибки и оперативно вносить изменения, снижая коммерческие риски [33 – 35].

Таким образом, в случае с разработкой программных средств просмотра фалов для ОС специального назначения, использование методологии RUP является наиболее рациональным, так как оно позволяет избежать существенных издержек еще на этапе проектирования, позволяет разложить всю работу по пунктам, что упрощает ее распределение между исполнителями и сокращает сроки выполнения проекта в общем.

Объектно-ориентированное программирование (ООП) является методологией программирования, в рамках которой программа представляется в виде совокупности объектов. Каждый из объектов является экземпляром определенного класса, обладающего определенным набором свойств и своим поведением. Классы объединены отношением наследования, что позволяет описывать новые классы на основе уже существующих, частично или целиком заимствовать их функциональность. Кроме того, важными свойствами ООП являются полиморфизм и инкапсуляция. Полиморфизм позволяет использовать объекты с одинаковым интерфейсом, не имея информации о типе и его внутренней структуре. Инкапсуляция позволяет скрывать данные и методы, работающие с ними, внутри класса.

1.6 Выбор программных средств

В качестве языка разработки приложения АОС был выбран язык Java, т.к. он является языком объектно-ориентированного программирования, позволяет проектировать сложные программные комплексы, при этом сохраняя оптимальную производительность. Кроме того, он является одним из двух языков, поддерживаемых средой разработки Android studio, предназначенной для создания мобильных приложений [36].

Для разработки программы формирования расширенной модели освоения учебных материалов был выбран объектно-ориентированный язык C#, т.к. он позволяет быстро писать эффективные программы под Windows. Для разработки интерфейса использовалась удобная технология WPF. В качестве среды программирования использовалась MS Visual studio 2019 Community [37, 38].

Язык онтологий SWRL – технология, основанная на объединении языков OWL и RuleML. В целом SWRL – технология, которая помогает описать абстрактный механизм оперирования субъектами предметной области, а также закономерности предметной области.

Язык онтологий OWL – это язык, служащий для определения и представления онтологий. Под онтологией понимают науку, описывающую формы бытия и то, как они относятся между собой. Сам термин был позаимствован из философии. Формальная семантика языка OWL описывает, как можно получить логические следствия, имея описанную онтологию, т.е. получить факты, которые не представлены в онтологии буквально, но выводятся из ее семантики. Эти следствия могут основываться на одном документе или множестве распределенных документов, комбинирующихся с использованием определенных механизмов.

Для разработки моделей и диаграмм на этапе анализа требований и проектирования будет использован программный продукт Draw.io. Данное веб-приложение поддерживает стандарт UML 3.0, легко в использовании и не

требует установки на компьютер [39, 40]. UML (англ. Unified Modeling Language – унифицированный язык моделирования) – это язык, служащий для графического описания объектного моделирования в области разработки программного обеспечения, а также моделирования бизнес-процессов, системного проектирования и отображения организации структур. Главным образом UML был разработан для помощи разработчикам в достижении соглашения в графических обозначениях для представления общих понятий (таких, как класс, компонент и т.д.) и уделять больше внимания проектированию архитектуры проекта [41 – 43].

Выводы

Анализ предметной области показал ряд слабых и сильных сторон современных АОС по физике. Для того, чтобы уменьшить негативные стороны и усилить позитивные, было принято решение разработать АОС с применением специально разработанной в рамках данной работы расширенной модели освоения учебного материала.

Также на основании актуальных данных была модифицирована классификация АОС.

Была рассмотрена модель освоения учебного материала Солова, выделены ее недостатки, которые были нивелированы разработанной на ее основе расширенной моделью освоения учебного материала.

Проведено сетевое планирование проекта аналитическим путем и с помощью среды MS Project, которое показало длительность реализации проекта – 81 день и предполагаемые затраты – 25040 руб. Из этого можно сделать вывод об экономической целесообразности проекта.

В конце был проведен и обоснован выбор технологий и средств разработки.

2 Разработка алгоритма формирования расширенной модели освоения учебного материала и проектирование АОС

2.1 Онтологическое моделирование

Проектирование АОС лучше всего проводить, опираясь на предварительный анализ, структуризацию и систематизацию информации предметной области. Прежде чем начинать разработку онтологической модели предметной области необходимо провести онтологический анализ [44, 45]. В ходе работы в среде Protege был разработан набор классов, соответствующий рассматриваемой задаче, свойств этих классов, а также набор связей и взаимодействий между ними. После, опираясь на них, была разработана на языках OWL и SWRL онтология в среде Protege. Одной из целей данной научно-исследовательской работы являлось построение онтологической модели предметной области. Список созданных классов представлен на рисунке 18. Общая схема модели представлена на рисунке 19.

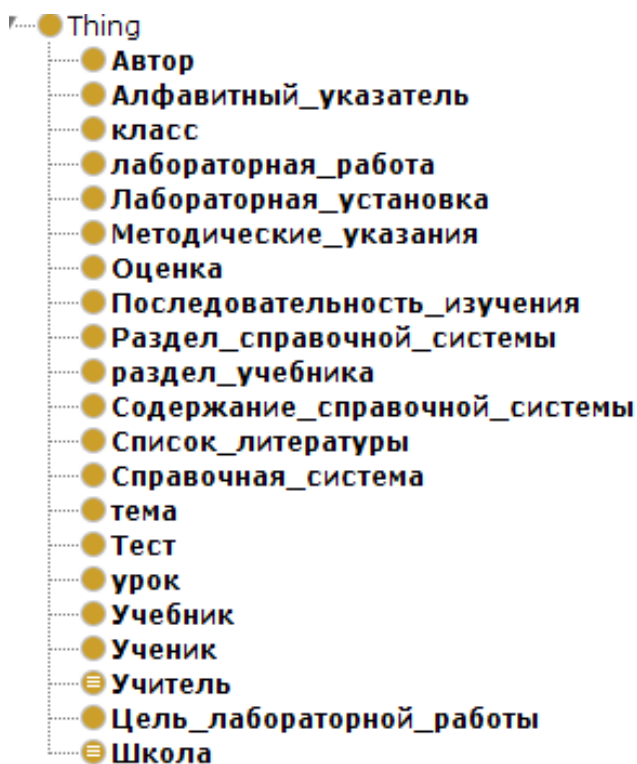


Рисунок 18 – Основной набор классов в дереве наследования

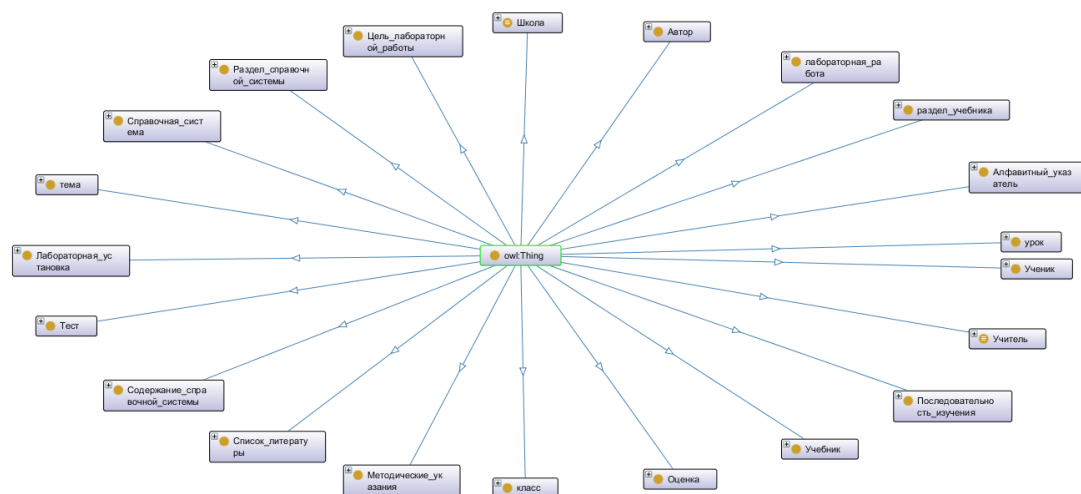


Рисунок 19 – Общий граф онтологической модели

Более подробные графы всех элементов этой онтологии и их связи поредставлена в приложении В.

Онтологическая модель была построена с помощью системы Protege и позволяет на базе структуризации и систематизации информации предметной области проводить эффективное проектирование АОС, в том числе на основе визуального моделирования с применением языка UML.

2.2 Анализ требований

2.2.1 Анализ функциональных требований

Главной задачей разрабатываемого приложения является обучение школьников различным физическим законам.

Благодаря проведению анализа предметной области были выделены следующие функции, которые должны быть реализованы в разрабатываемом приложении:

- проверка наличия у пользователя основного приложения и авторизации в нем;

- получение от сервера задания для лабораторной работы и списка задействованных в нем приборов;
- моделирование приборов и их физических свойств;
- регистрация действий пользователя;
- анализ результатов работы пользователя;
- предоставление справочной информации.

На рисунке 20 представлена диаграмма вариантов использования мобильного приложения, составленная на основе выделенных требований.

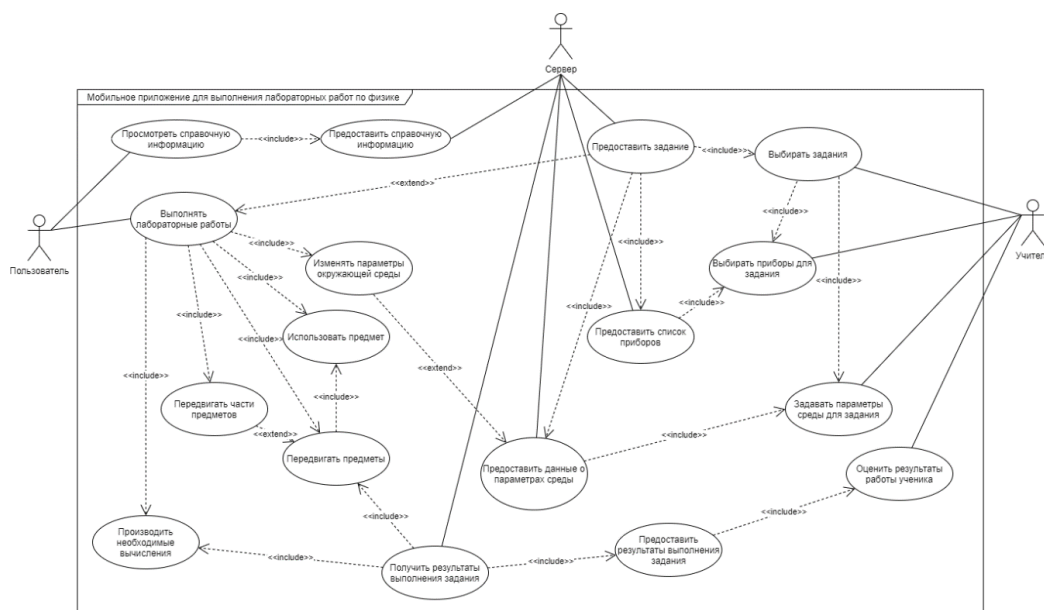


Рисунок 20 – Диаграмма вариантов использования мобильного приложения

Созданная диаграмма описывает создаваемую систему на концептуальном уровне. Было выделено 3 актера:

- ученик, выполняющий лабораторные работы в приложении;
- сервер, хранящий данные о возможных заданиях, лабораторных установках, используемых в них;
- учитель, выбирающий задания для учеников и задающий их параметры, а также проводящий проверку достижений учащихся.

Для подробного рассмотрения прецедентов приведем спецификацию прецедентов «Передвинуть предмет» и «Изменить параметры окружающей среды», которые изображены на рисунках 21 и 22 соответственно.

Прецедент: Передвинуть предмет
ID: 1
Краткое описание: Пользователь захватывает предмет и передвигает его
Главные актёры: Ученик
Второстепенные актёры: Нет
Предусловия: 1. Ученик загрузил приложение
Основной поток: 1. Ученик захватывает предмет 2. Ученик передвигает его на новые координаты 3. Система отрисовывает предмет по новым координатам 4. Если пользователь отпустил предмет 4.1 Система добавляет действие в список действий ученика 4.2 Система посылает этот список на сервер 4.3 Сервер получает запрос и добавляет данные в БД 4.4 Сервер отправляет ответ об удачном завершении 5. Иначе переход на действие 1
Постусловия: 1. Предмет передвинут
Альтернативные потоки: Нет.

Рисунок 21 – спецификация варианта использования «Передвинуть предмет»

Прецедент: Изменить параметры окружающей среды
ID: 2
Краткое описание: Пользователь задает новые значения параметрам окружающей среды
Главные актёры: Ученик
Второстепенные актёры: Нет
Предусловия: 2. Пользователь загрузил приложение
Основной поток: 1. Ученик нажимает кнопку, увеличивающую значение определенного параметра 2. Система сохраняет новое значение параметра 3. Регистрирует изменения в таблице и отображает его на экране
Постусловия: 1. Значение параметра среды изменено
Альтернативные потоки: Нет.

Рисунок 22 – спецификация варианта использования «Изменить параметры окружающей среды»

Для более наглядного рассмотрения требований к этим прецедентам, построим диаграммы деятельности для них. На рисунке 23 приведена диаграмма для варианта использования «Изменить параметры среды».

Теперь рассмотрим диаграмму деятельности для варианта использования «Передвинуть предмет» (рисунок 24).

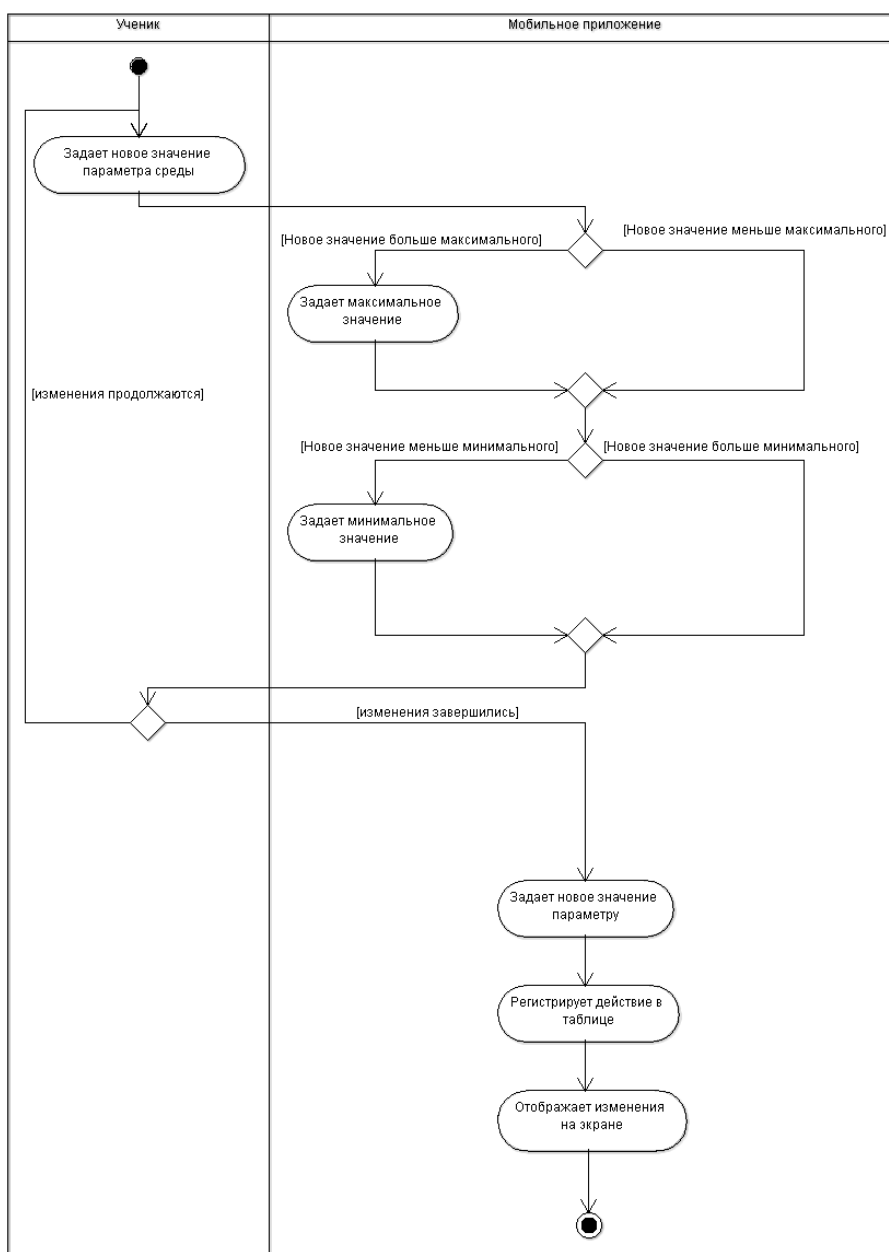


Рисунок 23 – Диаграмма деятельности для варианта использования «Изменить параметры среды»

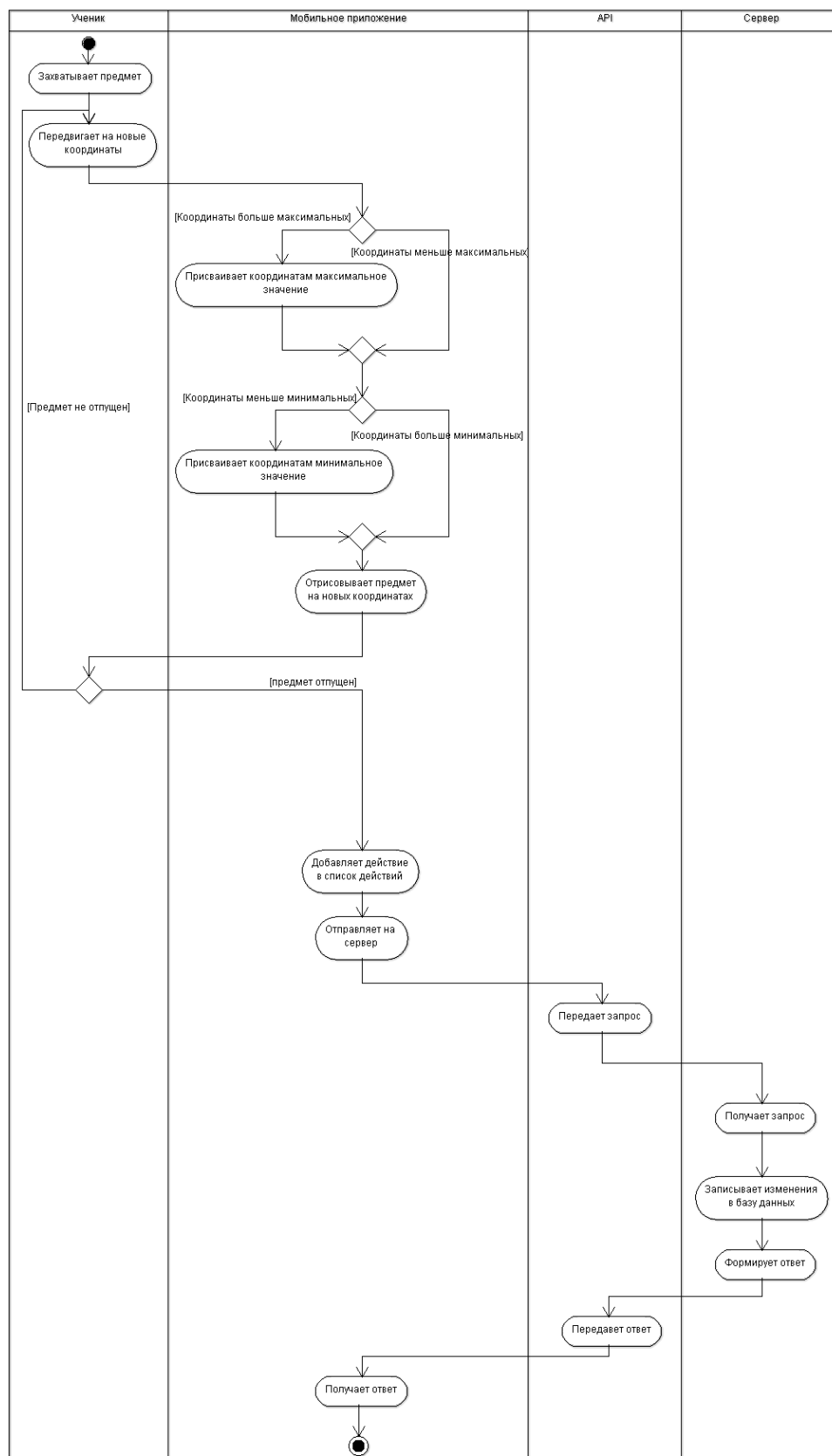


Рисунок 24 – Диаграмма деятельности для варианта использования «Передвинуть предмет»

Конечно, здесь были описаны требования не для всех вариантов использования приложения, но этого должно быть достаточно для примерного представления о функционале системы. Результаты проведенного анализа предметной области дают расширенное представление о протекающих в них процессах, а также позволять в дальнейшем разработать наиболее качественную и удобную программную систему.

2.2.2 Анализ нефункциональных требований

Помимо функциональных требований к приложению так же можно предъявить ряд нефункциональных требований, описанных далее в этом списке:

- его клиентская часть должна иметь интуитивно понятный интерфейс, с которым можно будет удобно взаимодействовать в режиме виртуальной реальности;
- необходимо, чтобы программа имела механизм защиты от отказов;
- приложение должно исполняться на мобильных устройствах под управлением операционной системы Android 8.0 «Oreo» и выше;
- передача данных между серверным и клиентским приложениями должна осуществляться в формате JSON-файла.

2.3 Разработка и исследование алгоритма формирования расширенной модели

2.3.1 Аналитическое исследование алгоритма с применением сетей Петри

Исследование программы формирования расширенной модели освоения учебного материала проводилось с использованием аппарата сетей Петри.

Вместе со своими многочисленными модификации, эта математическая модель является представителем класса моделей, одним из достоинств которых является возможность понятного представления не только структуры

сложных организационно-технологических систем и комплексов, но и логико-временных особенностей процессов их функционирования. Сеть Петри используется для представления структуры и анализа динамики функционирования систем в терминах «условие-событие». Эту модель успешно используют для описания динамических дискретных систем различных классов, к которым относятся: вычислительные процессы и программы, технологические процессы, информационные, экономические, биологические, социальные и технические системы [46 – 53].

Модели, построенные на основе сетей Петри, позволяют более эффективно исследовать работоспособность моделируемых систем, насколько оптимально они структурированы и эффективно функционируют, а также их возможность достижения во время выполнения тех или иных определенных состояний. Сети Петри являются удобным и мощным средством моделирования асинхронных, параллельных, распределенных и недетерминированных процессов, которые позволяют наглядно представить динамику функционирования систем и элементов, из которых они состоят. Свойство иерархического вложения сетей Петри позволяют рассматривать модели различной степени детализации, обеспечивая тем самым необходимую композицию сложных систем и процессов [46 – 53].

Различные свойства сетей Петри позволяют достаточно точно оценить работу алгоритма, а также позволяют обнаружить и устранить различные его недостатки, проявляемые в модели в виде тупиков, ловушек или недостижимых вершин.

На рисунке 25 представлена диаграмма деятельности программы, описывающая алгоритм формирования расширенной модели освоения учебного материала. В ней подробно расписаны все действия, которые необходимо выполнить пользователю и компьютеру для получения искомых графов.

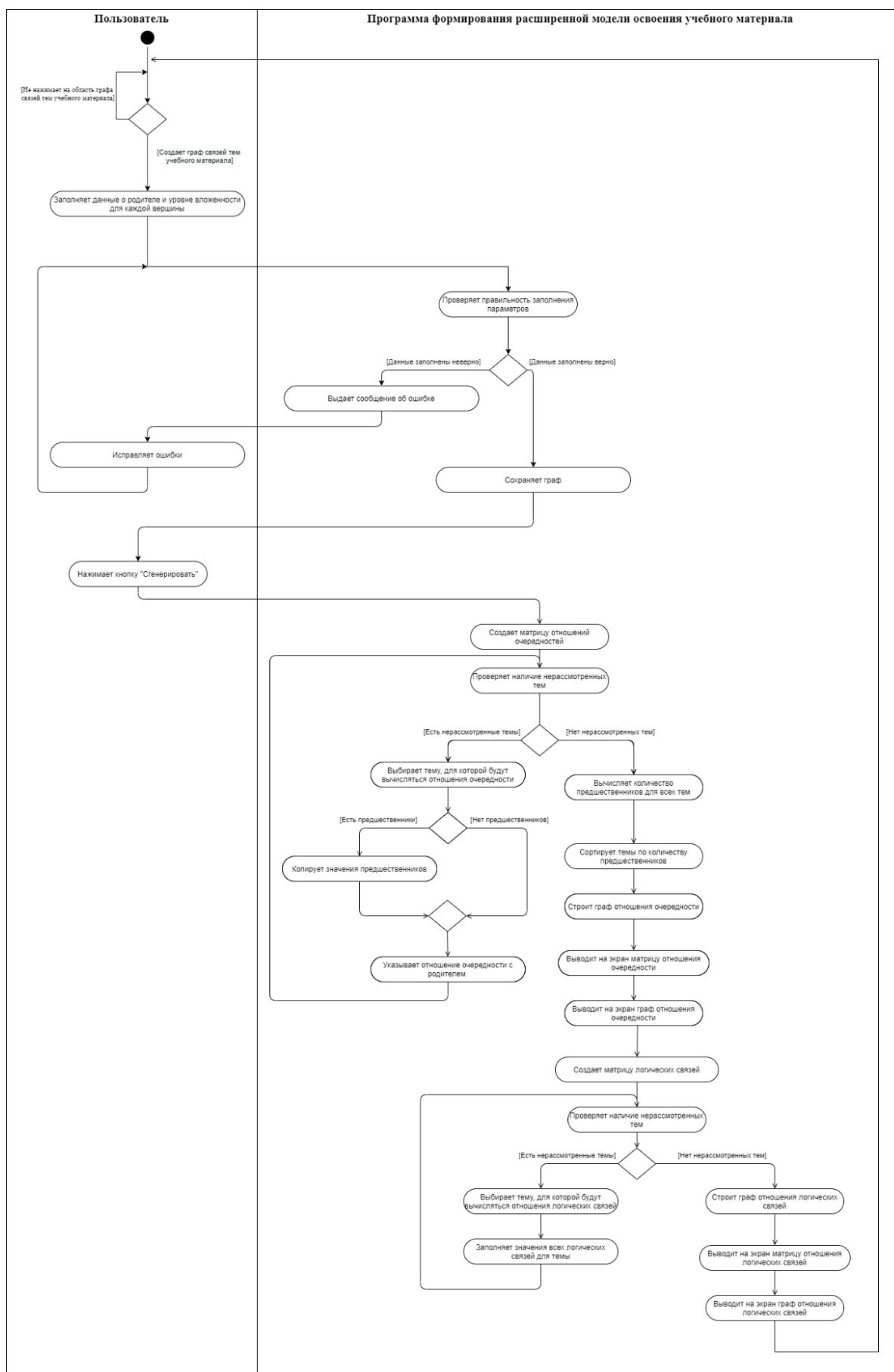


Рисунок 25 – Диаграмма деятельности алгоритма программы

На основе этой диаграммы можно построить сеть Петри, которая приведена на рисунке 26 [51, 53].

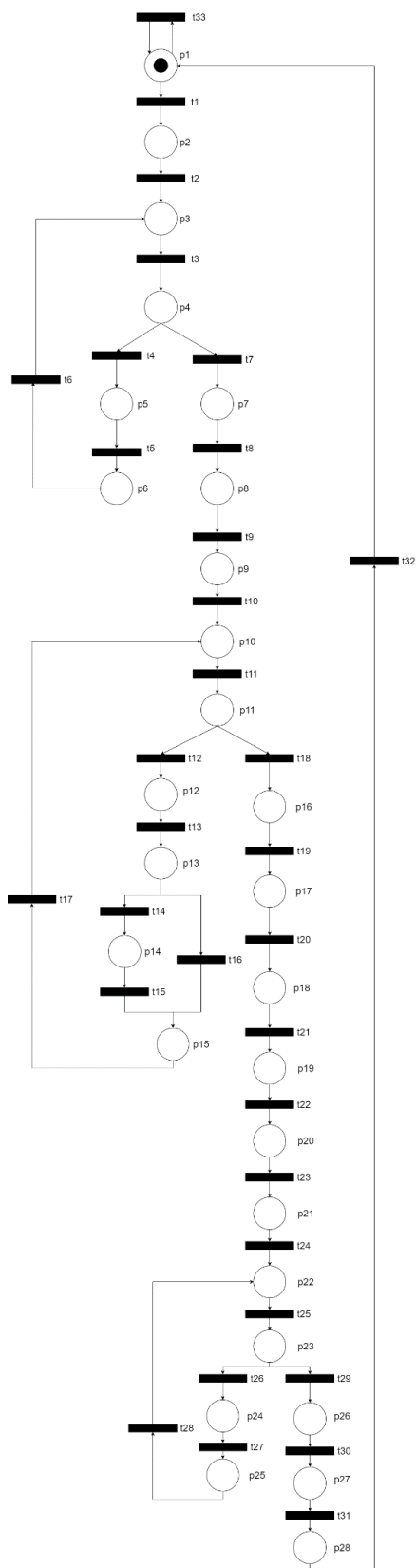


Рисунок 26 – Сеть Петри

В таблице 7 приведена интерпретация действий описываемой сети Петри.

Таблица 7 – Интерпретация действий сети Петри

Действие	Интерпретация
t1	Создает граф связей тем учебного материала
t2	Заполняет данные о родителе и уровне вложенности для каждой вершины
t3	Проверяет правильность заполнения параметров
t4	Данные заполнены неверно
t5	Выдает сообщение об ошибке
t6	Исправляет ошибки
t7	Данные заполнены верно
t8	Сохраняет граф
t9	Нажимает кнопку «сгенерировать»
t10	Создает матрицу отношений очередностей
t11	Проверяет наличие нерассмотренных тем
t12	Есть нерассмотренные темы
t13	Выбирает тему, для которой будут вычисляться отношения очередности
t14	Есть предшественники
t15	Копирует значения предшественников
t16	Нет предшественников
t17	Указывает отношение очередности с родителем
t18	Нет нерассмотренных тем
t19	Вычисляет количество предшественников для всех тем
t20	Сортирует темы по количеству предшественников
t21	Строит граф отношения очередности
t22	Выводит на экран матрицу отношения очередности
t23	Выводит на экран граф отношения очередности
t24	Создает матрицу логических связей
t25	Проверяет наличие нерассмотренных тем
t26	Есть нерассмотренные темы
t27	Выбирает тему, для которой будут вычисляться отношения логических связей
t28	Заполняет значения всех логических связей для темы
t29	Нет нерассмотренных тем
t30	Строит граф отношения логических связей
t31	Выводит на экран матрицу отношения логических связей
t32	Выводит на экран граф отношения логических связей
t33	Не нажимает на область графа связей тем учебного материала

Сеть Петри описывается следующими множествами:

$$C = (P, T, I, O).$$

Для исследуемой сети Петри множество состояний описывается следующим образом:

$$P = \{p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16, p17, p18, p19, p20, p21, p22, p23, p24, p25, p26, p27, p28\}.$$

Теперь опишем множество переходов:

$$T = \{t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13, t14, t15, t16, t17, t18, t19, t20, t21, t22, t23, t24, t25, t26, t27, t28, t29, t30, t31, t32, t33\}.$$

Позиция p_i является входной для перехода t_i в том случае, если $p_i \in I(t_i)$.

Позиция p_i является выходной для перехода t_i в том случае, если $p_i \in O(t_i)$ [47].

Определим расширенную входную и выходную функции для данной сети Петри:

$$I : P \rightarrow T^\infty, O : P \rightarrow T^\infty, I : T \rightarrow P^\infty, O : T \rightarrow P^\infty.$$

$$I(p1) = \{t32, t33\}$$

$$I(p16) = \{t18\}$$

$$I(p2) = \{t1\}$$

$$I(p17) = \{t19\}$$

$$I(p3) = \{t2, t6\}$$

$$I(p18) = \{t20\}$$

$$I(p4) = \{t3\}$$

$$I(p19) = \{t21\}$$

$$I(p5) = \{t4\}$$

$$I(p20) = \{t22\}$$

$$I(p6) = \{t5\}$$

$$I(p21) = \{t23\}$$

$$I(p7) = \{t7\}$$

$$I(p22) = \{t24, t28\}$$

$$I(p8) = \{t8\}$$

$$I(p23) = \{t25\}$$

$$I(p9) = \{t9\}$$

$$I(p24) = \{t26\}$$

$$I(p10) = \{t10, t17\}$$

$$I(p25) = \{t27\}$$

$$I(p11) = \{t11\}$$

$$I(p26) = \{t29\}$$

$$I(p12) = \{t12\}$$

$$I(p27) = \{t30\}$$

$$I(p13) = \{t13\}$$

$$I(p28) = \{t31\}$$

$$I(p14) = \{t14\}$$

$$I(p15) = \{t15, t16\}$$

$$O(p1) = \{t1, t33\}$$

$$O(p4) = \{t4, t7\}$$

$$O(p2) = \{t2\}$$

$$O(p5) = \{t5\}$$

$$O(p3) = \{t3\}$$

$$O(p6) = \{t6\}$$

$$O(p7) = \{t8\}$$

$$O(p8) = \{t9\}$$

$$O(p9) = \{t10\}$$

$$O(p10) = \{t11\}$$

$$O(p11) = \{t12, t18\}$$

$$O(p12) = \{t13\}$$

$$O(p13) = \{t14, t16\}$$

$$O(p14) = \{t15\}$$

$$O(p15) = \{t17\}$$

$$O(p16) = \{t19\}$$

$$O(p17) = \{t20\}$$

$$O(p18) = \{t21\}$$

$$O(p19) = \{t22\}$$

$$O(p20) = \{t23\}$$

$$O(p21) = \{t24\}$$

$$O(p22) = \{t25\}$$

$$O(p23) = \{t26, t29\}$$

$$O(p24) = \{t27\}$$

$$O(p25) = \{t28\}$$

$$O(p26) = \{t30\}$$

$$O(p27) = \{t31\}$$

$$O(p28) = \{t32\}$$

$$I(t1) = \{p1\}$$

$$I(t2) = \{p2\}$$

$$I(t3) = \{p3\}$$

$$I(t4) = \{p4\}$$

$$I(t5) = \{p5\}$$

$$I(t6) = \{p6\}$$

$$I(t7) = \{p4\}$$

$$I(t8) = \{p7\}$$

$$I(t9) = \{p8\}$$

$$I(t10) = \{p9\}$$

$$I(t11) = \{p10\}$$

$$I(t12) = \{p11\}$$

$$I(t13) = \{p12\}$$

$$I(t14) = \{p13\}$$

$$I(t15) = \{p14\}$$

$$I(t16) = \{p13\}$$

$$I(t17) = \{p15\}$$

$$I(t18) = \{p11\}$$

$$I(t19) = \{p16\}$$

$$I(t20) = \{p17\}$$

$$I(t21) = \{p18\}$$

$$I(t22) = \{p19\}$$

$$I(t23) = \{p20\}$$

$$I(t24) = \{p21\}$$

$$I(t25) = \{p22\}$$

$$I(t26) = \{p23\}$$

$$I(t27) = \{p24\}$$

$$I(t28) = \{p25\}$$

$$I(t29) = \{p23\}$$

$$I(t30) = \{p26\}$$

$$I(t31) = \{p27\}$$

$$I(t32) = \{p28\}$$

$$I(t33) = \{p1\}$$

$$O(t1) = \{p2\}$$

$$O(t2) = \{p3\}$$

$$O(t3) = \{p4\}$$

$$O(t4) = \{p5\}$$

$$O(t5) = \{p6\}$$

$$O(t6) = \{p3\}$$

$$O(t7) = \{p7\}$$

$$O(t8) = \{p8\}$$

$$O(t9) = \{p8\}$$

$$O(t10) = \{p10\}$$

$$O(t11) = \{p11\}$$

$$O(t12) = \{p12\}$$

$$O(t13) = \{p13\}$$

$$O(t14) = \{p14\}$$

$$O(t15) = \{p15\}$$

$$O(t16) = \{p15\}$$

$$O(t17) = \{p10\}$$

$$O(t18) = \{p16\}$$

$$O(t19) = \{p17\}$$

$$O(t20) = \{p18\}$$

$$O(t21) = \{p19\}$$

$$O(t22) = \{p20\}$$

$$O(t23) = \{p21\}$$

$$O(t24) = \{p22\}$$

$$O(t25) = \{p23\}$$

$$O(t26) = \{p24\}$$

$$O(t27) = \{p25\}$$

$$O(t28) = \{p22\}$$

$$O(t29) = \{p26\}$$

$$O(t30) = \{p27\}$$

$$O(t31) = \{p28\}$$

$$O(t32) = \{p1\}$$

$$O(t33) = \{p1\}$$

$$\mu = (1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0).$$
[illegible]

Рисунок 27 – Дерево достижимости для сети Петри

Рассмотрим свойства данной сети Петри.

Ограниченность (или K -ограниченность) имеет место, если число фишек в любой позиции сети не может превысить значения K ($K=1$). Исходя из этого условия, данная сеть является ограниченной [47 – 53].

Одним из важнейших свойств сети Петри является безопасность. Безопасность – это частный случай ограниченности, а именно это 1-ограниченность. Позиция сети Петри является безопасной, если число фишек в ней никогда не превышает 1. Сеть Петри безопасна, если безопасны все ее позиции. Построив дерево достижимости, мы убедились, что в каждой позиции нашей сети количество фишек не превышает 1, следовательно, сеть является безопасной.

Переходы сети Петри можно разбить на категории по уровню активности следующим образом [47 – 52]:

а) уровень 0 – переход t_j обладает активностью уровня 0, если он никогда не может быть запущен;

б) уровень 1 – переход t_j обладает активностью уровня 1, если он потенциально запустим, т. е. если существует такая μ , $R(C, \mu)$, что t_j разрешен в μ' ;

в) уровень 2 – переход t_j обладает активностью уровня 2, если для всякого целого n существует последовательность запусков, в которой t_j присутствует по крайней мере n раз;

г) уровень 3 – переход t_j обладает активностью уровня 3, если существует бесконечная последовательность запусков, в которой t_j присутствует неограниченно часто;

д) уровень 4 – переход t_j обладает активностью уровня 4, если для всякой μ' $R(C, \mu)$ существует такая последовательность запусков σ , что t_j разрешен в $\delta(\mu, \sigma)$.

В ходе анализа данной сети все переходы находятся на одном уровне активности – 4.

Сеть Петри должна обладать свойством сохранения. Для этого фишки, не должны создаваться или уничтожаться, общее число фишек в сети должно оставаться постоянным. Строгое сохранение требует, чтобы число входов в каждый переход должно равняться числу выходов $|I(t_j)| = |O(t_j)|$. Данная сеть является сохраняющей.

2.3.2 Эксперимент в программе Petri net emulator

После ручных расчетов дерева достижимости сети Петри была осуществлена их проверка путем проведения эксперимента в программе Petri net emulator. На рисунках 28 и 29 представлена введенная в программу сеть Петри.

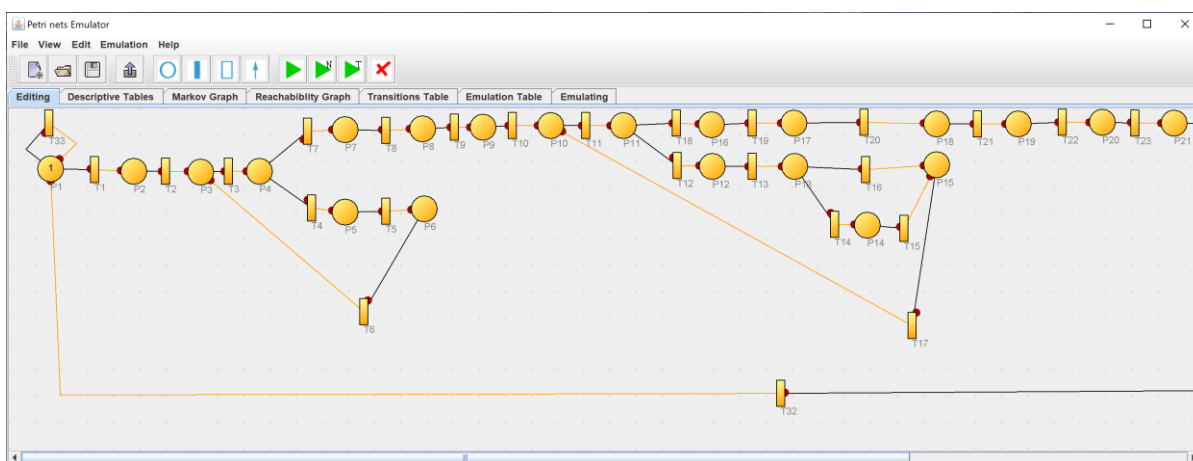


Рисунок 28 – Граф сети Петри (часть 1)

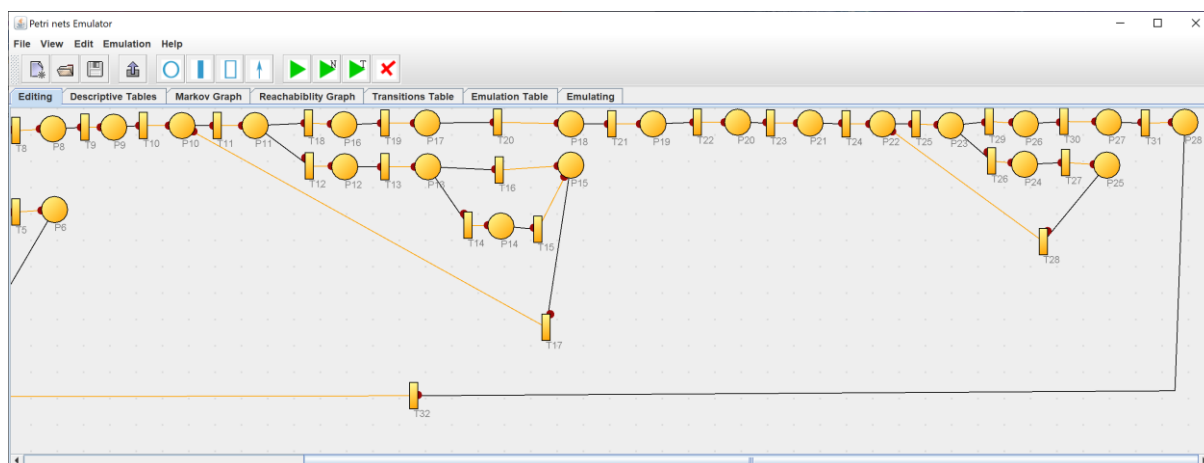


Рисунок 29 – Граф сети Петри (часть 2)

Далее приведем построенное программой дерево достижимости в табличной (рисунок 30) и графической (рисунки 31 и 32) форме. Представленные программой выводы совпадают с полученными при ручном расчете.

На рисунках 33 и 34 представлены результаты симуляции в табличной форме.

[illegible]

Рисунок 33 – Результаты симуляции (часть 1)

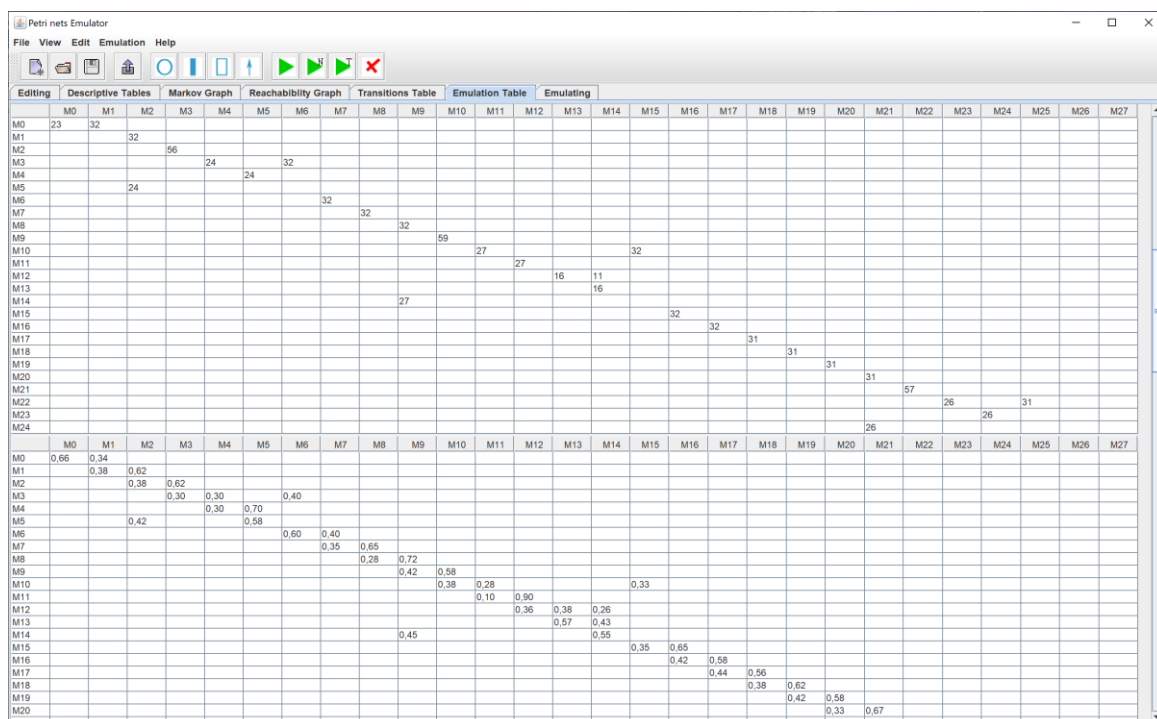


Рисунок 34 – Результаты симуляции (часть 2)

На рисунках 35 и 36 представлены полученные в ходе программы гистограммы, отображающие работу сети Петри.

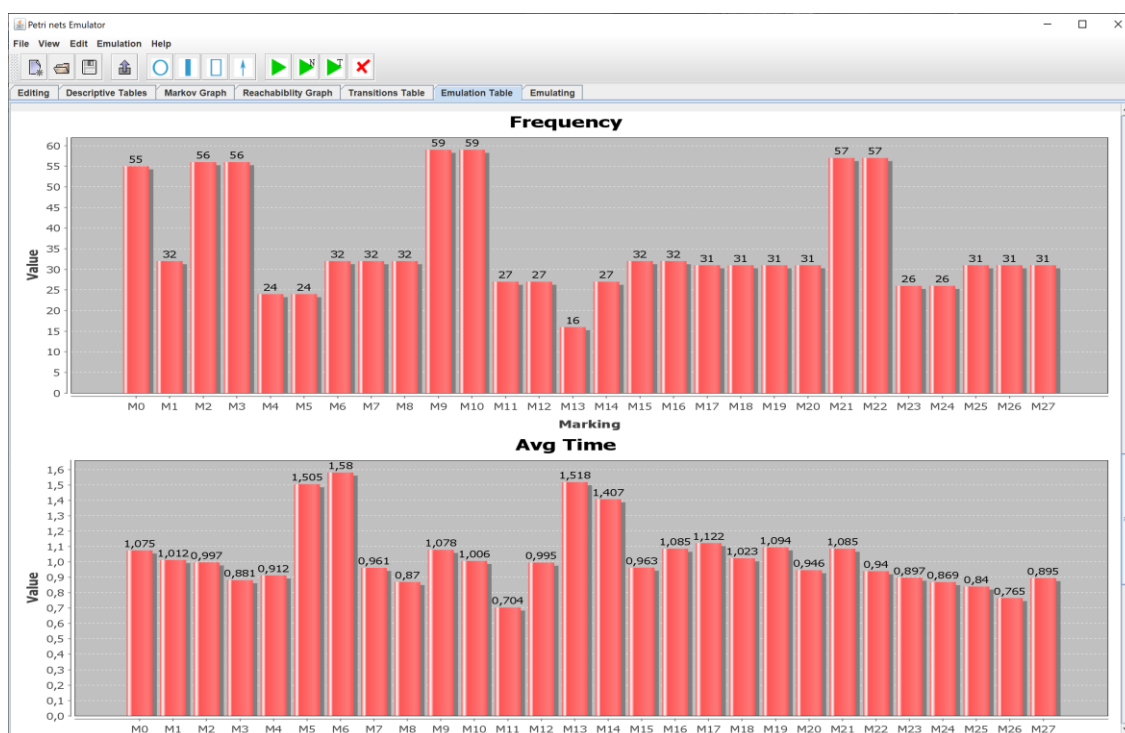


Рисунок 35 – Гистограммы (часть 1)

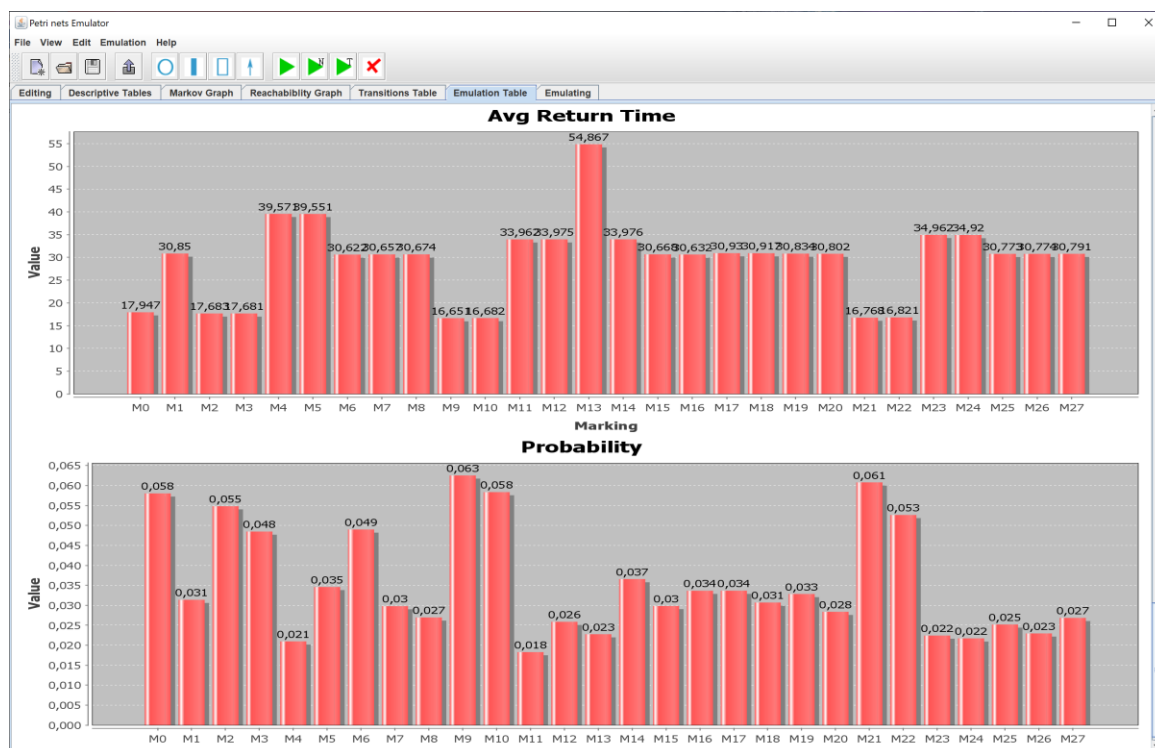


Рисунок 36 – Гистограммы (часть 2)

Проведённый анализ построенной сети Петри подтвердил наличие свойства безопасности, ограниченности, отсутствие тупиков и ловушек. Следовательно, разработанный алгоритм программы формирования расширенной модели освоения учебного материала является корректным и надежным.

2.4 Разработка интерфейса приложения

2.4.1 Разработка интерфейса программы формирования расширенной модели учебного материала

Для автоматизации процесса формирования расширенной модели учебного материала была написана программа, выполняющая этот процесс. На вход программе подается исходный граф, построенный с помощью встроенного конструктора. По нему программа формирует 2 других – граф последовательности учебного материала и граф логических связей. Первый определяет рациональную последовательность, в которой необходимо

подавать учебный материала для лучшего его усвоения. Второй в наглядной форме показывает все логические взаимосвязи между учебными элементами. Скриншоты интерфейса программы приведены на рисунках 37 – 40.

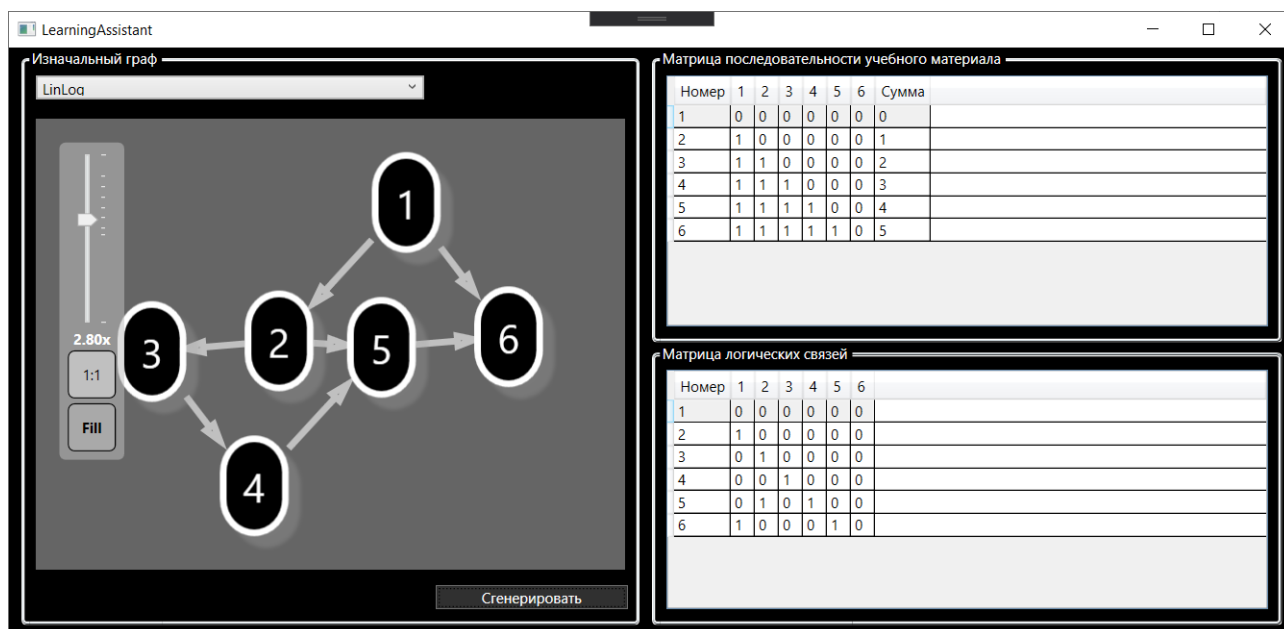


Рисунок 37 – Основное окно программы с примером графа

The screenshot shows a dialog box titled 'Добавить элемент' (Add element). It contains four input fields with the following labels and values:

- Название (Name): 7
- Порядковый номер (Sequential number): 7
- Номер уровня (Level number): 2
- Родитель (Parent): 6

At the bottom of the dialog box, there are two buttons: 'Отмена' (Cancel) and 'Сохранить' (Save).

Рисунок 38 – Окно добавления вершины

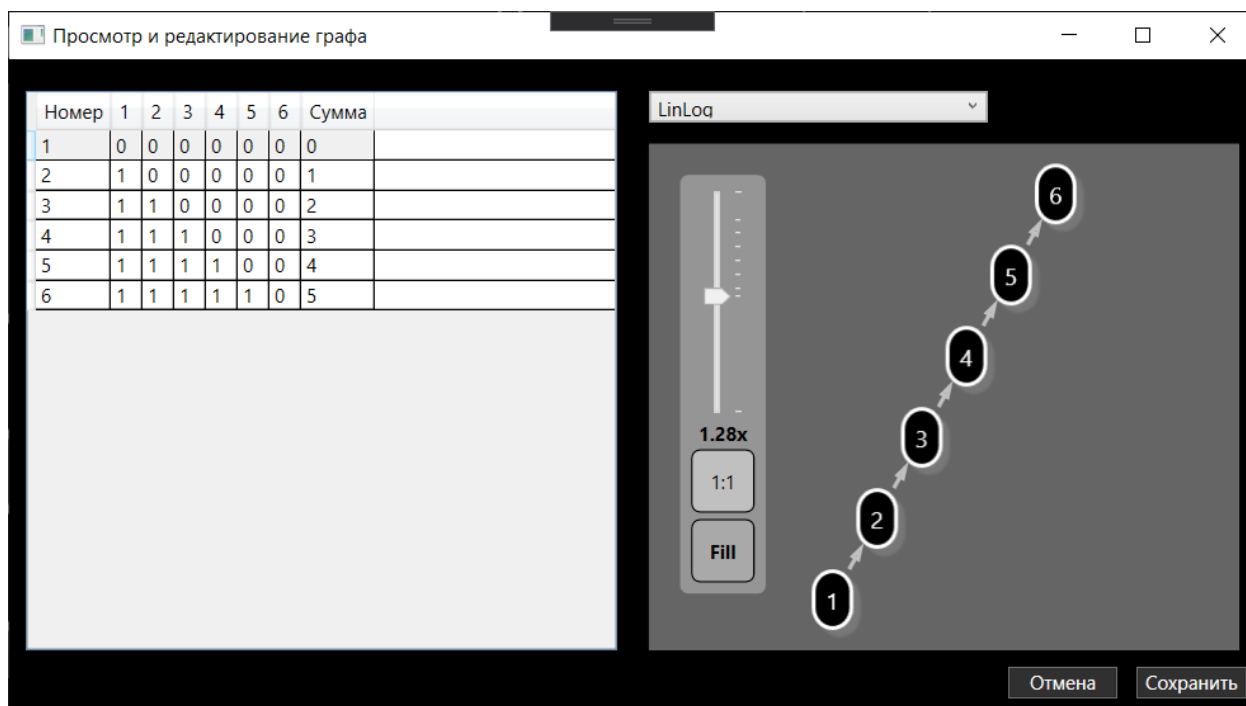


Рисунок 39 – Окно просмотра графа последовательности

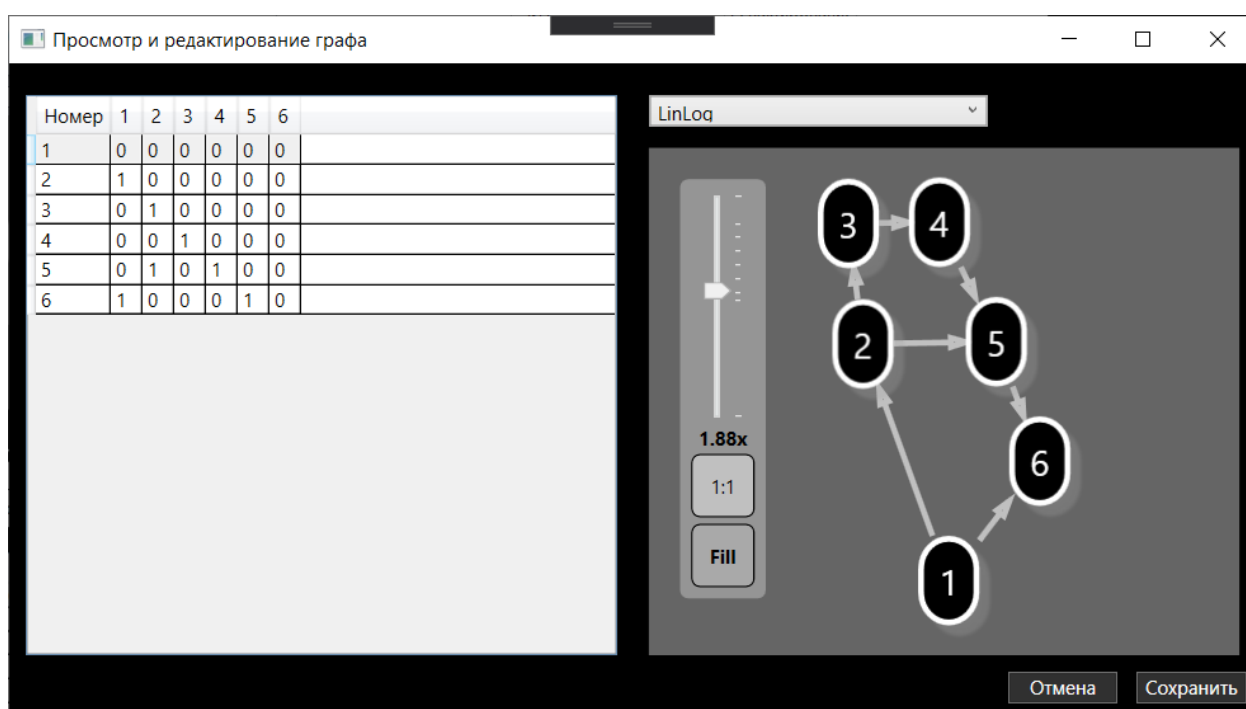


Рисунок 40 – Окно просмотра и редактирования графа логических связей

Помимо проблемы разработки расширенной модели освоения учебных материалов необходимо так же рассмотреть еще одну – визуализация работы лабораторных установок, в частности математического маятника.

Существующие примеры алгоритмов для данного вида маятника, обычно, проигрывают по одному из этих показателей. Необходимо разработать такое решение, которое было бы компромиссным и давало достаточно точный результат, затрачивая при этом минимальное время.

2.4.2 Разработка интерфейса АОС

В рамках проектирования АОС и мобильного приложения, помимо визуального моделирования, проведено проектирование соответствующего графического интерфейса, приведенного в приложении Г.

Разработанный интерфейс позволяет удобно переключаться между разделами физики, а сама структура материала разработана на основе полученного программой формирования расширенной модели освоения материала графа.

Выводы

Проведены формирование и анализ функциональных и нефункциональных требований для разрабатываемой АОС, проведено онтологическое моделирование рассматриваемой предметной области.

Разработан алгоритм формирования расширенной модели освоения учебного материала и спроектирована программа, реализующая алгоритм его формирования. Также этот алгоритм был протестирован с использованием аппарата сетей Петри, что показало его корректность и надежность.

Затем, на основании результатов анализа, был разработан интерфейс для Программы формирования расширенной модели освоения учебного материала и разрабатываемой АОС.

3 Реализация и тестирование

3.1 Структура программных средств

Разработка приложения осуществлялась на языке программирования Java в среде Android Studio.

В приложении Б исходный код справочной системы АОС. Исходный код Программы формирования расширенной модели освоения учебного материала выложен на github [54].

На рисунках 41 и 42 приведены диаграммы классов, отражающие общую структуру исходного кода программы.

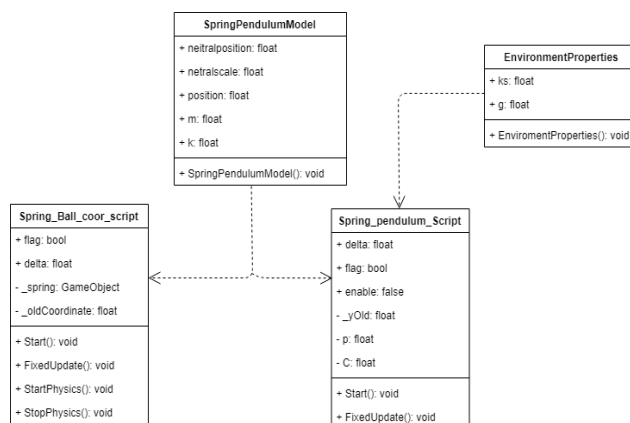


Рисунок 41 – Диаграмма классов для прибора «пружинный маятник»

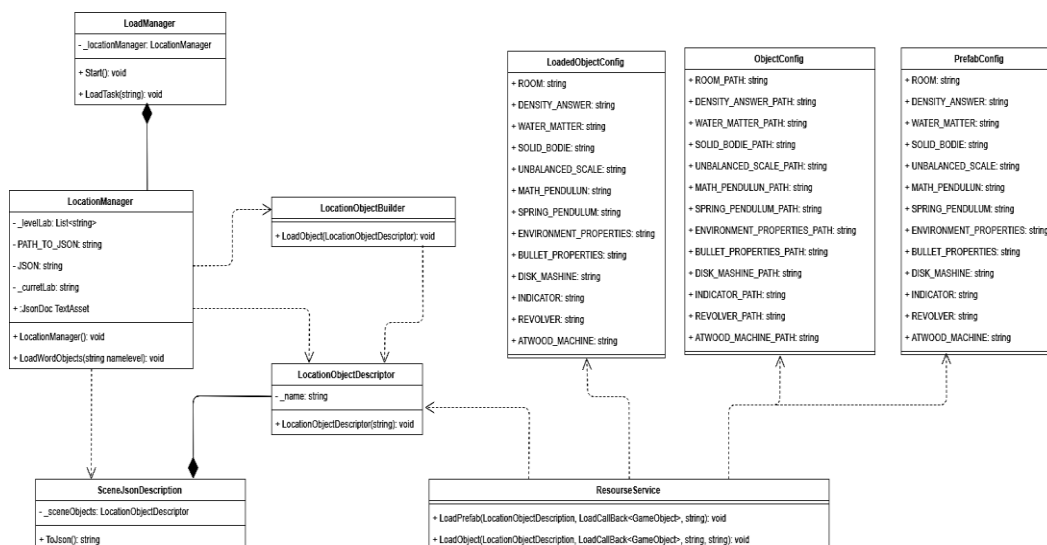


Рисунок 42 – Диаграмма классов для загрузки задания и объектов на сцену

На рисунке 43 представлена диаграмма развертывания АОС.

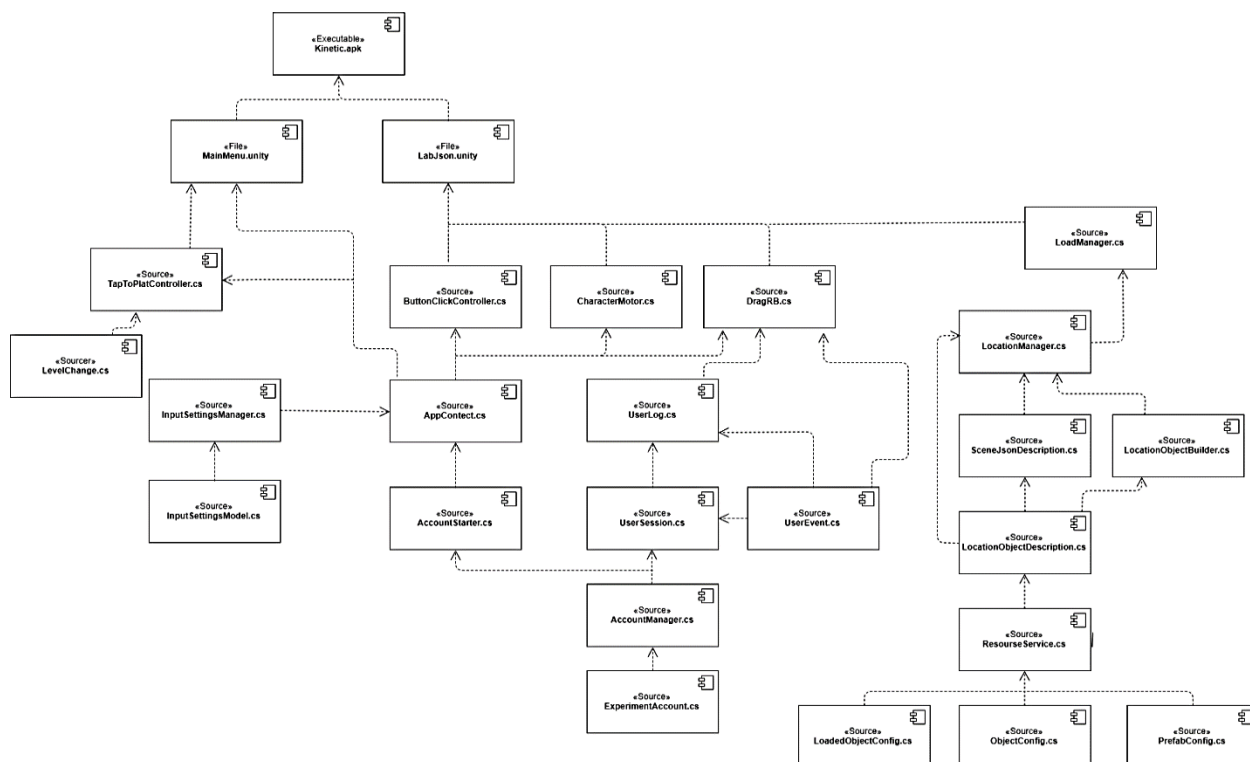


Рисунок 43 – Диаграмма классов для загрузки задания и объектов на сцену

На рисунке 44 представлена диаграмма развертывания, отображающая распределение компонентов приложения по физическим узлам во время работы приложения.

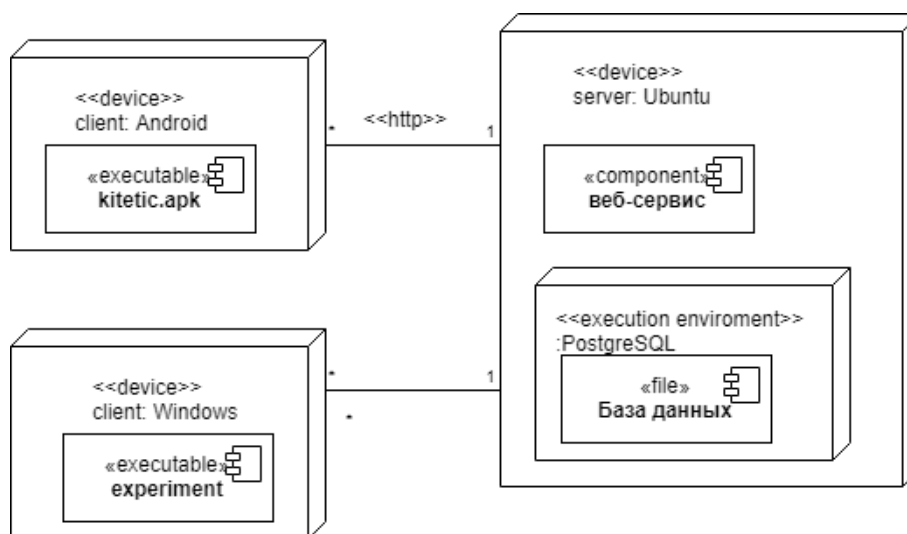


Рисунок 44 – Диаграмма развертывания

На этом рисунке видно, что клиентская часть приложения, разработка которой было целью выпускной квалификационной работы, работает на мобильных устройствах под управлением операционной системы Android. Во время работы приложение связывается с серверной частью, развернутой на компьютере под управлением операционной системы Ubuntu, на котором так же расположена база данных, созданная на базе СУБД PostgreSQL, которая хранит данные, полученные из другого клиентского приложения (web-приложения), в котором учитель задает задания.

3.2 Тестирование программного продукта

3.2.1 Тестирование программы формирования расширенной модели освоения учебного материала

Для вычисления объектно-ориентированных метрик программного кода магистерской диссертации на тему «Исследование и разработка автоматизированных обучающих систем с использованием технологии дополненной реальности» был выбран класс LogicalMatrix Программы формирования расширенной модели освоения учебного материала, отвечающий за логическое представление матрицы логических связей учебных материалов [55].

3.2.1.1 Объектно-ориентированные метрики

Для оценки качества программы для нее были рассчитаны следующие объектно-ориентированные метрики: WMC, DIT, RFC, сцепление и NMS.

WMC (Weighted methods per Class) – количество «взвешенных» методов на класс, с помощью которого можно измерить сложность классов. В качестве оценки сложности методов будем использовать количество строк текста, которое они занимают (L_i , где i – номер метода). В классе LogicalMatrix

определено 4 метода, имеющие соответственно следующие сложности: 5, 8, 15, 5. Тогда

$$WMC(LogicalMatrix) = L_1 + L_2 + L_3 + L_4 = 5 + 8 + 15 + 5 = 33.$$

DIT (Depth of Inheritance Tree) – глубина дерева наследования, позволяет определить количество классов-предков, потенциально оказывающих влияние на данный класс. Рассматриваемый класс `LogicalMatrix` имеет один родительский класс – абстрактный класс `GraphMatrix`, который, в свою очередь, как и все классы в `C#`, наследуется от класса `Object`. Таким образом:

$$DIT(LogicalMatrix) = 2.$$

RFC (Response for a class, число реакций на класс) – $RFC = |RS|$, где RS – ответное множество класса, то есть множество методов, которые могут быть потенциально вызваны методом класса в ответ на данные, полученные объектом класса. То есть $RS = ((\{M\}(\{R_i\})), i=1...n$, где M – все возможные методы класса, R_i – все возможные методы, которые могут быть вызваны i -м классом. Тогда RFC будет являться мощностью данного множества. Чем больше RFC, тем сложнее тестирование и отладка.

Ответное множество класса служит индикатором слишком сильной связанности классов в проекте, поэтому данная метрика особенно актуальна для масштабных систем.

Число методов класса = 4. Тогда получаем:

$$RFC = \sum_{i=0}^3 RSi = 3 + 4 + 4 + 1 = 12.$$

Некоторые методы этого класса имеют зависимости от одних и тех же методов, из-за чего полученное число RFC может несколько завышено. В любом случае, оно не выходит за пределы нормы для систем с количеством классов, аналогичным рассматриваемой.

Теперь просчитаем сцепление. В данную категорию входят 2 метрики: C_a (центростремительное сцепление) и C_e (центробежное сцепление).

Вычислим центростремительное сцепление, определяющее количество классов вне этой категории, которые зависят от классов внутри этой категории.

Количество классов, зависящих от `LogicalMatrix`, составляет 2.

Затем просчитаем центробежное сцепление. Для определяется количество классов внутри категории, которые зависят от классов вне ее.

Количество классов, от которых зависит `LogicalMatrix`, составляет 6.

Данные метрики также показывают степень связанности классов в проекте. Для класса `LogicalMatrix` 2 и 6 являются небольшими значениями.

Последняя метрика – NMS (Number of Message-Sends, Количество сообщений, посылаемых данным методом).

С помощью данной метрики оценим степень коммуникационной сложности тех же четырех методов.

Для метода `AddGraphEdges`:

$$NMS(AddGraphEdges) = 2.$$

Для метода `CreateGraphEdges`:

$$NMS(CreateGraphEdges) = 2.$$

Для метода `CalculatePriorityGraph`:

$$NMS(CalculatePriorityGraph) = 3.$$

Для метода `CalculateMatrix`:

$$NMS(CalculateMatrix) = 1.$$

Класс `LogicalMatrix` редко обращается к другим классам, в основном, другие классы обращаются к нему.

С помощью подсчета объектно-ориентированных метрик качества кода было установлено, что программный код проекта имеет достаточный уровень тестируемости и поддерживаемости.

3.2.1.2 Метрики Лоренса и Кидда

Были просчитаны следующие метрики Лоренса и Кидда: NKC и NSUB.

NKC (Number of Key Classes) – это количество ключевых классов.

Ключевой класс прямо связан с коммерческой проблемной областью, для которой предназначена система. Рекомендуемое значение ключевых классов составляет не менее 20% и не более 40%.

Общее количество классов = 20.

Количество ключевых классов = 6.

В данной программе ключевыми являются классы, отвечающие за хранение и обработку данных о графах отношений учебных материалов:

$$NKC = 6/24 = 0.25 = 25\%.$$

Таким образом, система подходит под оптимальное значение данной метрики.

NSUB (Number of SUBsystem) – определяется количеством подсистем.

Количество подсистем обеспечивает понимание следующих вопросов: размещение ресурсов, планирование (с акцентом на параллельную разработку), общие затраты на интеграцию.

Рекомендуемое значение: NSUB больше 3.

Система по управлению жизненным циклом активных правил включает в себя следующие подсистемы:

- GUI с использованием WPF;
- клиентская часть;
- основная подсистема для работы с активными правилами.

Количество подсистем находится ровно на границе рекомендованного Лоренсом и Киддом значения.

3.2.1.3 Встроенные метрики кода MS Visual Studio

Microsoft предоставляет встроенное в Visual Studio средство, которое позволяет оценить качество программного кода проекта. Среда разработки подсчитывает следующие 5 метрик кода:

- индекс удобства поддержки;
- сложность организации циклов;
- глубина наследования;
- взаимосвязь классов;
- количество строк кода.

Результаты подсчета представлены на рисунках 45 и 46.

Иерархия	Индекс удобства поддержки	Сложность организации ц...	Глубина наследования	Взаимозависимость классов	Строки исходного кода	Строки исполняемого кода
LearningAssistant (Debug)	83	204	9	93	1732	458
LearningAssistant	73	65	9	70	663	152
LearningAssistant.EducationalMaterialGraphBits	94	16	2	4	71	6
EducationalMaterialEdge	94	3	2	2	17	1
EducationalMaterialGraph	97	3	2	3	10	0
EducationalMaterialVertex	92	10	1	0	35	5
LearningAssistant.GraphCalculators	64	13	2	14	85	26
PriorityMatrix	64	13	2	14	82	26
LearningAssistant.Matrix	85	47	2	29	300	62
GraphMatrix	100	4	1	4	17	0
IGraphMatrixViewController	100	3	0	2	6	0
LogicalMatrix	64	11	2	13	72	27
LogicalMatrixViewController	72	6	1	15	45	11
LogicalMatrixViewData	83	1	2	2	9	2
MatrixRowData	90	6	1	3	38	6
MatrixViewData	100	7	1	2	21	0
PriorityMatrixViewController	69	6	1	15	51	13
PriorityMatrixViewData	90	3	2	2	18	3
LearningAssistant.RelationshipsGraphBits	94	12	2	4	54	4
RelationshipsEdge	94	3	2	2	16	1
RelationshipsGraph	97	3	2	3	10	0
RelationshipsVertex	93	6	1	0	19	3
LearningAssistant.ViewModels	83	33	9	11	399	167
EducationalMaterialGraphLayout	100	1	9	4	5	0
EducationalMaterialGraphViewModel	69	9	1	4	124	36
LogicalRelationshipsGraphViewModel	64	10	2	7	118	64
PriorityViewModel	66	11	2	7	126	67

Рисунок 45 – Метрики кода Visual Studio (часть 1)

Иерархия	Индекс удобства поддержки	Сложность организации ц...	Глубина наследования	Взаимозависимость классов	Строки исходного кода	Строки исполняемого кода
EducationalMaterialVertex	92	10	1	0	35	5
LearningAssistant.GraphCalculators	64	13	2	14	85	26
PriorityMatrix	64	13	2	14	82	26
LearningAssistant.Matrix	85	47	2	29	300	62
GraphMatrix	100	4	1	4	17	0
IGraphMatrixViewController	100	3	0	2	6	0
LogicalMatrix	64	11	2	13	72	27
LogicalMatrixViewController	72	6	1	15	45	11
LogicalMatrixViewData	83	1	2	2	9	2
MatrixRowData	90	6	1	3	38	6
MatrixViewData	100	7	1	2	21	0
PriorityMatrixViewController	69	6	1	15	51	13
PriorityMatrixViewData	90	3	2	2	18	3
LearningAssistant.RelationshipsGraphBits	94	12	2	4	54	4
RelationshipsEdge	94	3	2	2	16	1
RelationshipsGraph	97	3	2	3	10	0
RelationshipsVertex	93	6	1	0	19	3
LearningAssistant.ViewModels	83	33	9	11	399	167
EducationalMaterialGraphLayout	100	1	9	4	5	0
EducationalMaterialGraphViewModel	69	9	1	4	124	36
LogicalRelationshipsGraphViewModel	64	10	2	7	118	64
PriorityViewModel	66	11	2	7	126	67
RelationshipsGraphLayout	100	1	9	4	5	0
ViewModel	100	1	1	0	4	0
LearningAssistant.Windows	71	13	9	20	160	32
AddElementWindow	71	13	9	20	153	32
XamlGeneratedNamespace	87	5	2	13	0	9

Рисунок 46 – Метрики кода Visual Studio (часть 2)

Зеленые индикаторы означают, что код соответствует требованиям, прописанным компанией Microsoft.

Таким образом, на основе рассчитанных метрик можно сделать вывод, что программа формирования расширенной модели освоения учебного материала спроектирована и разработана в соответствии с существующими стандартами качества кода, что позволяет без затруднений добавлять в систему новый функционал и покрывать ее тестами, а также делает код понятным для последующих разработчиков.

3.2.2 Тестирование АОС

Важной частью жизненного цикла любого программного обеспечения является его тестирование, главная цель которого не только проверка на наличие ошибок, но и проверка соответствия созданного продукта предъявленным к нему требованиям [56, 57]. Поэтому представленный в этой работе программный продукт тоже был подвергнут этому процессу.

3.2.2.1 Функциональное тестирование

Рассмотрим данный вид тестирования на примере метода использования «Задание пользователем параметров окружающей среды», а конкретно параметров «угол отклонения математического маятника» и «ускорение свободного падения» [56].

Техническое задание для данной функции можно сформировать так: распознать заданные пользователем значения угла отклонения математического маятника и ускорения свободного падения и скорректировать их в случае ошибок ввода.

Тестируемый метод должен анализировать указанные пользователем угол отклонения маятника и ускорение свободного падения и запускать его движение, если параметры указаны верно. Иначе подгонять их под ближайшие возможные значения и также запускать движение.

3.2.2.1.1 Метод эквивалентного разбиения

Данная функция имеет следующие ограничения входных данных:

- угол отклонения больше минус 90° ;
- угол отклонения меньше 90° ;
- ускорение свободного падения меньше машинного максимума.

Классы эквивалентности, входящие в них входные данные, а также варианты реагирования системы на них приведены в таблице 8.

Таблица 8 – Классы эквивалентности

Входные данные	Класс эквивалентности	Выходные данные	Полученный результат
Угол отклонения $> -90^\circ$	Правильно заданный угол маятника		Маятник запущен
Угол отклонения $< 90^\circ$			
Угол отклонения $> 90^\circ$	Неправильно заданный угол маятника		Система выставляет максимально (минимально) возможный угол
Угол отклонения $< -90^\circ$			
Ускорение свободного падения > 0	Правильно заданный параметр ускорение свободного паления		Параметр установлен
Ускорение свободного падения $< \text{Машинного максимума}$			
Ускорение свободного падения < 0	Неправильно заданный параметр ускорение свободного паления		Система выставляет минимально возможное значение

Благодаря этой таблице можно сделать вывод, что для каждого параметра существуют только 2 класса эквивалентности: правильные и

неправильные значения. В случае первых система должна просто запустить движения маятника. Во втором случае ей необходимо сначала подогнать значения параметра под нужные пределы и только потом запускать движение.

3.2.2.1.2 Метод границ

Данный метод тестирования лучше всего рассмотреть на примере параметра длина нити маятника. Возможные для него ограничения показаны на рисунке 47, на котором символом x обозначается длина установки с зажимом, на которой висит нить. Сами границы представлены в таблице 9.

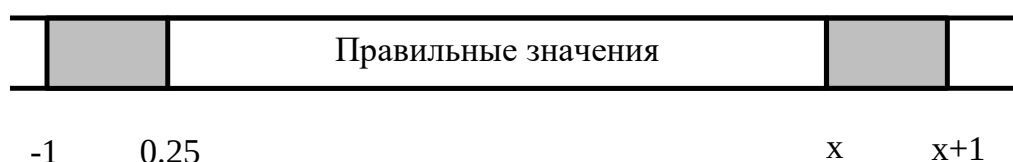


Рисунок 47 – Диапазон значений параметра длина нити

Таблица 9 – Границы параметра «длина нити»

Тип границ	Входные данные	Выходные данные	Полученный результат	Ошибка
Произвольные (выход за пределы длины установки с зажимом)	8,5	Параметр длины принял значение 8.5	Отрисовка нити длины 8,5	
	x	Параметр длины принял значение x	Отрисовка нити длины x	
	$x + 1$	Параметр длины принял значение $x + 1$	Отрисовка нити длины x	
	0,25	Параметр длины принял значение 0,25	Отрисовка нити длины 0,25	

Продолжение таблицы 9

Тип границ	Входные данные	Выходные данные	Полученный результат	Ошибка
Физические	-1	Параметр длины принял значение 0,25	Отрисовка нити длины 0,25	
	0	Параметр длины принял значение 0,25	Отрисовка нити длины 0,25	
Технологические (выход за пределы float)	-1,5 $\times 10^{-45}$	Параметр длины принял значение 0,25	Отрисовка нити длины 0,25	
	1,5 $\times 10^{-45}$	Параметр длины принял значение x	Отрисовка нити длины x	

Границы можно разделить на 3 группы: произвольные (выход за пределы длины установки с зажимом), физические (не может быть меньше 0), и технологические (выход за пределы float). Для всех видов границ система должна проверить выход длины нити маятника за их пределы и выставить граничные значения, если это произошло.

3.2.2.1.3 Метод функциональных диаграмм

Данный метод будет проверять правильность распознавания системой всех параметров вместе и их совокупное влияние на запуск движения маятника.

На рисунке 48 представлена функциональная диаграмма, на которой мы видим, что для запуска движения маятника все параметры должны быть указаны верно.

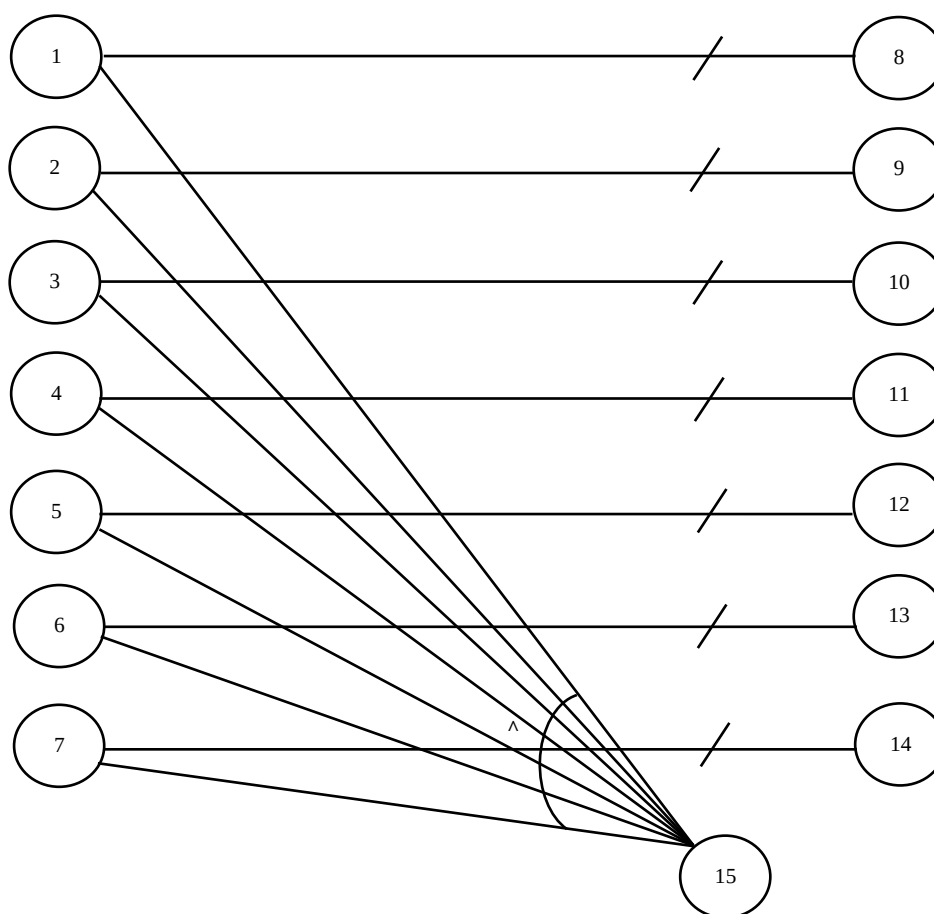


Рисунок 48 – Функциональная диаграмма

Цифры, использованные на этом рисунке, имеют следующие значения:

- 1 – длина нити маятника больше 0;
- 2 – длина нити маятника меньше длины установки с зажимом;
- 3 – масса груза больше 0;
- 4 – ускорение свободного падения больше 0;

- 5 – плотность среды больше юольше 0;
- 6 – угол отклонения больше минус 90°;
- 7 - угол отклонения меньше 90°;
- 8 – установлено минимальное значение длины нити;
- 9 – установлено максимальное значение длины нити;
- 10 – установлено минимальное значение массы груза;
- 11 – установлено минимальное значение ускорения свободного падения;
- 12 – установлено минимальное значение плотности среды;
- 13 – установлено минимальное значение угла;
- 14 – установлено максимальное значение угла;
- 15 – начато движение маятника.

В таблице 10, построенной на основе предыдущей диаграммы, указаны решения, на основе которых уже непосредственно строятся тесты, представленные в таблице 11.

Таблица 10 – Таблица решений

	1	2	3	4	5	6	7	8
1	0							1
2		0						1
3			0					1
4				0				1
5					0			1
6						0		1
7							0	1
8	1	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0
11	0	0	0	1	0	0	0	0
12	0	0	0	0	1	0	0	0
13	0	0	0	0	0	1	0	0
14	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	1

Таблица 11 – Тесты, полученные в результате применения функционального подхода

[illegible]

Продолжение таблицы 11

№ теста		Длина нити маятника меньше 0	Длина нити маятника больше длины установки	Масса груза меньше нуля	Ускорение свободного падения меньше 0	Плотность среды меньше 0	Угол меньше -90°	Угол больше 90°	Ошибок в параметрах нет
6	Угол меньше -90°	Установлено минимальное значение угла	Установлено минимальное значение угла	Установлено минимальное значение угла	Установлено минимальное значение угла	Установлено минимальное значение угла	Установлено минимальное значение угла	Установлено минимальное значение угла	-
7	Угол больше 90°	Установлено максимальное значение угла	Установлено максимальное значение угла	Установлено максимальное значение угла	Установлено максимальное значение угла	Установлено максимальное значение угла	Установлено максимальное значение угла	Установлено максимальное значение угла	-
8	Ошибок в параметрах нет	-	-	-	-	-	-	-	Начато движение маятника

Благодаря применению этого метода тестирования получилось минимизировать количество тестов для проверки работы системы со всеми параметрами, потому что он помог выявить самые важные зависимости параметров друг от друга, что позволило избежать прямого перебора всех возможных комбинаций их значений.

3.2.2.2 Метрики кода

3.2.2.2.1 Цикломатическая метрика Мак-Кейба

Эта метрика основывается на анализе графа программы (см. рисунок 9) и определяет цикломатическую сложность программы $V(G)$.

Для ее определения сначала нужно подсчитать на графе количество дуг (e):

$$e = 24.$$

После этого просчитывается количество вершин (n):

$$n = 19.$$

Затем нужно определить число компонент связи (p) – количество дуг, которое необходимо добавить, чтобы из каждой вершины графа можно было прийти в любую другую вершину. Для рассматриваемого случая

$$p = 1.$$

Подставляет полученные данные в формулу цикломатической сложности и получаем:

$$V(G) = e - n + 2p = 7.$$

Из полученного значения метрики можно сделать вывод, что рассматриваемый фрагмент программы является достаточно понятным, хоть и не самым простым.

3.2.2.2.2 Метрики кода, вычисленные с помощью Android Studio

Также был произведен расчет метрик в среде разработки Android Studio, с помощью использования плагина Metrics Reloaded. Результаты представлены на рисунке 49.

Metric Name	Value	Status
Cyclomatic Complexity	7	Red
Lines of Code	19	Yellow
...	...	...

Рисунок 49 – Метрики кода, вычисленные с помощью плагина Metrics Reloaded для Android Studio

Из них видно, что код имеет простоту обслуживания выше средней и неплохие показатели сложности организации, глубины наследования, и количества связанных классов, что свидетельствует о достаточно высоком уровне качества программного кода.

Выводы

В рамках реализации были разработаны диаграммы классов и компонентов. Реализована программа формирования расширенной модели учебного материала и мобильное приложение для АОС на языках высокого уровня C# и Java соответственно. Оба приложения были протестированы различными способами. Результаты тестирования показали, что обе программы соответствуют возложенным на них требованиям.

Полученные результаты, связанные с разработкой мобильного приложения для АОС по школьному курсу физики, в частности лабораторный практикум, используются в рамках образовательной онлайн-платформы Experiment, выполняемой IT-компанией ООО «СЕТАП».

Заключение

В ходе выполнения диссертационной получены следующие основные результаты:

а) предложен и исследован аналитически и экспериментально с помощью математического аппарата сетей Петри модифицированный алгоритм формирования расширенной модели освоения учебного материала, отличающийся от известных тем, что позволяет комплексно учитывать его иерархическую структуру и эффективно реализовывать варианты траекторий изучения материала при построении гипертекста АОС;

б) предложена расширенная онтологическая модель предметной области для АОС, отличающаяся от известных более углубленным анализом с целью формализации и систематизации информации для эффективного проведения проектирования объекта исследования;

в) предложена модифицированная классификация АОС, отличающаяся от известных тем, что предлагается более полный расширенный перечень классификационных признаков и систем;

г) проведен анализ предметной области и требований, проанализированы существующие аналоги, выявлены их преимущества и недостатки, разработаны функциональные и нефункциональные требования к создаваемому программному продукту;

д) было проведено планирование разработки и оценка бюджета программного проекта;

е) было проведено проектирование компонентов программного проекта разработки АОС;

ж) были проведены реализация и тестирование компонентов программного проекта разработки АОС;

и) по материалам диссертации:

- опубликованы 7 научных статей,
- сделаны доклады на 5-и научных конференциях,

- получены 2 диплома лауреата 1-й степени и 2 грамоты за победы в номинациях Международных конкурсов научных работ,
- получено Свидетельство о государственной регистрации программы для ЭВМ.

Таким образом, с помощью анализа и исследования моделей освоения учебных материалов и алгоритмов их формирования был разработан программный продукт, полностью удовлетворяющий установленным требованиям.

Список использованных источников

1. Распоряжение Президента Российской Федерации от 9 мая 2017 г. № 203 «Стратегия развития информационного общества в Российской Федерации на 2017 – 2030 годы». [Электронный ресурс]. URL: http://zmedu.hostedu.ru/files/ykaz_7668.pdf (дата обращения: 16.09.2019).
2. Программа «Цифровая экономика Российской Федерации». Утверждена распоряжением Правительства Российской Федерации от 28 июля 2017 г. № 1632-р. [Электронный ресурс]. URL: <http://static.government.ru/media/files/9gFM4FHj4PsB79I5v7yLVuPgu4bvR7M0.pdf> (дата обращения: 16.09.2019).
3. Федеральный закон «Об образовании в Российской Федерации» № 273 – ФЗ от 29.12.2012 года с изменениями 2019 года. [Электронный ресурс]. – URL: <http://zakon-ob-obrazovanii.ru/> (дата обращения: 16.09.2019).
4. Христочевский, С.А. Базовые элементы электронных учебников и мультимедийных энциклопедий. / С.А. Христочевский. – М.: Вики, 2009. – 497 с.
5. Тулупова, Т. В. Автоматизация проектирования обучающих систем / Т.В. Тулупова. – Институт Государственного управления, права и инновационных технологий (ИГУПИТ) Интернет-журнал «Науковедение» №5 2013.
6. Храмцова, Е.О. Интеллектуальные обучающие системы / Храмцова Е.О., Бочкарев П.В. // Международный научно-технический журнал «Теория. Практика. Инновации» декабрь 2017.
7. Пескова, Е.С. Повышение эффективности профессиональной подготовки бакалавров технических университетов на основе применения электронно-образовательных ресурсов / Е.С. Пескова. – Томск 2015.
8. Кудинов, Д.Н. Перспективы разработки автоматизированных обучающих систем. / Д.Н. Кудинов. – СПб.: Проспект Науки, 2008. – 310 с.

9. Аязбаев, Т.Л. Технология создания компьютерных обучающих программ. / Аязбаев Т.Л., Галагузова Т.А. // Международный журнал экспериментального образования. – 2015. - № 3 (часть 1). – с. 76-78.

10. Алексеев, Г.В. Основы разработки электронных учебных изданий. / Г.В. Алексеев, И.И. Бриденко, Е.И. Верболоз – СПб.: Проспект Науки, 2010. – 450 с.

11. Лабораторный практикум по физике института дистанционного образования. – [Электронный ресурс]. URL: <https://ido.tsu.ru/schools/physmat/data/res/virtlab/> (дата обращения: 19.12.2019).

12. Онлайн справочник по физике All-fizika.com. – [Электронный ресурс]. URL: <http://www.all-fizika.com/> (дата обращения: 19.12.2019).

13. Страница приложения «Физика от А до Я, помощник ЕГЭ». – [Электронный ресурс]. URL: <https://play.google.com/store/apps/details?id=ru.trainer.physics2&hl=ru> (дата обращения: 19.12.2019).

14. Джонатан, Л. Виртуальная реальность в Unity. / Л. Джонатан – М.: ДМК Пресс, 2016 – 316 с.

15. Андрушко, Д.Ю. Применение технологий виртуальной и дополненной реальности в образовательном процессе: проблемы и перспективы / Д.Ю. Андрушко – Научное обозрение. – 2018. - № 6. – с. 5-10.

16. Мишина, А.В. Реализация основных дидактических принципов обучения в процессе применения систем дополненной реальности при создании обучающих систем иностранному языку для детей дошкольного возраста / Мишина А.В. // Интернет-журнал «Мир науки» 2016, Том 4, номер 3.

17. Иванько, А.Ф. Дополненная и виртуальная реальность в образовании / Иванько А.Ф., Иванько М.А., Бурцева М.Б. // Молодой ученый. – 2018. - № 37 (223). с. 11-16.

18. Гаврилова, Е. Г. Технологии дополненной реальности и другие «сказочные» возможности в работе библиотеки / Е. Г. Гаврилова.

19. Фоминых, М. И. Учебно-исследовательский проект Живой учебник по информатике с дополненной реальностью / М. И. Фоминых. – Ирбит, 2018.

20. Суворов, К.А. Системы виртуальной реальности и их применение / Суворов К.А. // Т-Comm – Телекоммуникации и Транспорт – 2013, – с.140 – 143

21. Акулич, М. Дополненная, виртуальная, смешанная реальность и маркетинг / М. Акулич. – «Издательские решения».

22. Луценко, Е.В. Блеск и нищета виртуальной реальности / Луценко Е.В. // Научный журнал КубГАУ – 2016 – #124(10)

23. Кяргин, А.Ю. Приложение виртуальной реальности по курсу химии с использованием системы распознавания жестов / Кяргин А.Ю., Кирилловых А.М., Цветковская И.И. // Виртуальная и дополненная реальность – 2016: состояние и перспективы / Сборник научно-методических материалов, тезисов и статей конференции. Под общей редакцией д.т.н., проф. Д.И. Попова. – М.: Изд-во ГПБОУ МГОК, 2016. – 386 с.

24. Осипов, М.П. ТЗ8 Системы виртуальной реальности. Учебно-методическое пособие / М.П. Осипов. – Нижний Новгород: Нижегородский госуниверситет, 2012 – 48 с.

25. Соловов, А.В. Проектирование компьютерных систем учебного назначения: Учебное пособие. – [Электронный ресурс]. URL: <http://window.edu.ru/resource/488/3488> (дата обращения: 16.09.2019).

26. Акчурина, В.Д. Разработка автоматизированной обучающей системы с использованием технологии дополненной реальности / Акчурина В.Д., Князев В.Н. // Сборник научных статей XII Международного молодежного конкурса научных работ «Молодежь в науке: новые аргументы». Липецк: Научное партнерство «Аргумент» – 2019 – с. 59-68.

27. Акчурина, В.Д. Разработка автоматизированной обучающей системы с использованием технологии дополненной реальности / Акчурина В.Д., Князев В.Н. // Сборник научных статей XII Международного молодежного конкурса научных работ «Молодежь в науке: новые аргументы». Липецк: Научное партнерство «Аргумент» – 2019, – с. 59-68.

28. Акчурина, В.Д. Вопросы разработки автоматизированной обучающей системы по школьному курсу физики с использованием технологии дополненной реальности / Акчурина В.Д., Князев В.Н. // Сборник избранных статей по материалам Международной научной конференции: Технические и естественные науки. – СПб.: ГНИИ «Нацразвитие» – 2020, – с. 74-79.

29. Акчурина, В.Д. Вопросы обучения курсу физики с использованием технологии дополненной реальности / Акчурина В.Д., Князев В.Н. // Сборник избранных статей по материалам Международной научной конференции : Наука. Исследования. Практика. Часть 2. СПб.: ГНИИ «Нацразвитие» – 2020, – с. 114-119.

30. Акчурина, В.Д. Вопросы разработки автоматизированной обучающей системы по курсу физики / Акчурина В.Д., Князев В.Н. // Сборник научных статей VII Всероссийской межвузовской научно-практической конференции «Информационные технологии в науке и образовании. Проблемы и перспективы» Пенза – 2020, – с. 121-123.

31. Акчурина, В.Д. Вопросы создания автоматизированной обучающей системы для школьного курса физики с применением технологии дополненной реальности / Акчурина В.Д., Князев В.Н. // Сборник материалов Международной научно-практической конференции «Актуальные проблемы науки и образования в Условиях современных вызовов». Москва, ООО «ИРОК», 2021. с. 33-40.

32. Акчурина В.Д. Разработка автоматизированной обучающей системы с использованием технологии дополненной реальности / Акчурина В.Д.,

Князев В.Н. // Международный научно-исследовательский журнал ч.1 #7 (109), 2021.

33. Орлов, С. А. Технологии разработки программного обеспечения / С. А. Орлов, Б.Я. Цилькер. – СПб: Питер, 2012. – 464 с.

34. Кролл, П. Rational Unified Process – это легко. Руководство по RUP для практиков / П. Кролл, Ф. Крачтен. – М.: КУДИЦ-ОБРАЗ, 2004. – 432 с.

35. Крачтен, Ф. Введение в Rational Unified Process. / Ф. Крачтен. – М.: Вильямс, 2002. – 462 с.

36. Официальный сайт Android Studio – [Электронный ресурс]. URL: <https://developer.android.com/studio> (дата обращения: 12.06.2021).

37. Полное руководство по языку программирования C# 7.0 и платформе .NET 4.7 [Электронный ресурс]. URL: <https://metanit.com/sharp/tutorial/> (дата обращения 29.09.2020).

38. Официальный сайт Visual Studio – [Электронный ресурс]. URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 12.06.2021).

39. Обзор инструмента draw.io – [Электронный ресурс]. URL: <https://startpack.ru/application/draw-io> (дата обращения: 12.06.2021).

40. Инструмент draw.io – [Электронный ресурс]. URL: <https://app.diagrams.net/> (дата обращения: 12.06.2021).

41. Арлоу, Д. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Д. Арлоу, А. Нейштадт. – СПб.: Символ Плюс, 2007. – 624 с.

42. Шмуллер, Д. Освой самостоятельно UML за 24 часа / Д.Шмуллер. – М.: Вильямс 2009. – 352с.

43. Креативные решения в проектировании, бизнесе и в частной жизни. UML диаграммы. [Электронный ресурс]. URL: planerka.info (дата обращения: 04.05.21).

44. Цуканова, Н.И. Онтологическая модель представления и организации знаний. Учебное пособие для вузов. / Н.И. Цуканова. – М.: Горячая линия – Телеком, 2015.

45. Палагин, А.В. Онтологические методы и средства обработки предметных знаний / А.В. Палагин, С.Л. Кривый, Н.Г. Петренко. – Луганск: изд-во ВНУ им. В. Даля, 2012. – 324 с.

46. Котов, В.Е. Сети Петри / В.Е. Котов. – М.: Наука, 1984. – 160 с.

47. Питерсон, Дж. Теория сетей Петри и моделирование систем / Дж. Питерсон. – М.: Мир, 1984. – 264с.

48. Проститенко, О. В. Моделирование дискретных систем на основе сетей Петри: учебное пособие / О.В. Проститенко, В.И. Халимон, А.Ю. Рогов. – СПб.: СПбГТИ(ТУ), 2017. – 69 с.

49. Веретельникова, Е.Л. Теоретическая информатика. Теория сетей Петри и моделирование систем: учебное пособие / Е.Л. Веретельникова. – Новосибирск: Изд-во НГТУ, 2018. – 82 с.

50. Лескин, А.А. Сети Петри в моделировании и управлении / А.А. Лескин, П. А. Мальцев, А. М. Спиридонов. – Л.: Наука, 1989. – 133с.

51. Ишакова, Е.Н. Теория вычислительных процессов: учебное пособие / Е.Н. Ишакова. – Оренбург: ГОУ ОГУ, 2007. – 160 с.

52. Зубова, Т.Н. Использование сетей Петри для моделирования процесса принятия управленческих решений / Зубова Т.Н., Мельников Б.Ф. // Вектор науки ТГУ, №3 (17), 2011. с. 33-37.

53. Кудж, С.А. Моделирование с использованием сетей Петри / Кудж С.А., Логинова А.С. // Вестник МГТУ МИРЭА 2015 №1 (6), 2015. с. 10-22.

54. Код Программы формирования расширенной модели освоения учебного материала на GitHub. [Электронный ресурс]. URL: <https://github.com/akchurinabarbara/LearningAssistant> (дата обращения: 24.06.21).

55. Романов, В.Ю. Анализ программного обеспечения с использованием объектно-ориентированных метрик. Обзор метрик./ Романов В.Ю. // International Journal of Open Information Technologies, том 2, номер 3, 2014

56. Функциональное тестирование. [Электронный ресурс]. URL: <http://tpl-it.wikispaces.com> (дата обращения: 29.04.21).

57. Майерс, Г. Искусство тестирования программ / Г. Майерс; пер. с англ. под ред. Б. А. Позина. – М.: Фиансы и статистика, 1982. – 176 с.

Приложение А
ГЛОССАРИЙ
(справочное)

Автоматизированная информационная система (АИС) – это совокупность различных программно-аппаратных средств, которые предназначены для автоматизации какой-либо деятельности, связанной с передачей, хранением и обработкой различной информации.

UML (Unified Modeling Language) – это язык визуального описания для проведения объектного моделирования в области разработки ПО.

Android – операционная система для смартфонов, планшетов, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, и других устройств. В настоящее время большинство смартфонов мира функционируют именно на этой ОС. Первая версия Android вышла в 2008 году, вместе с первым полноценным пакетом для разработки приложений SDK 1.0.

Rational Unified Process – одна из методологий разработки программного обеспечения, разработанная компанией Rational Software. Она использует итеративную модель разработки. В конце каждой итерации (продолжительность которых в идеальном случае составляет от 2 до 6 недель) проектная команда должна достичь запланированных целей, создать или доработать проектные артефакты и получить промежуточную, но функциональную версию конечного продукта.

Механика – раздел физики, в котором рассматривается движение материальных тел и взаимодействие между ними.

Кинетика – микроскопическая теория процессов в неравновесных средах.

Математический маятник – осциллятор, представляющий собой механическую систему, состоящую из материальной точки на конце невесомой нерастяжимой нити или лёгкого стержня и находящуюся в однородном поле сил тяготения.

Приложение Б

ИСХОДНЫЙ КОД ПРОГРАММЫ

(рекомендуемое)

Листинг Б.1 – Код класса MainActivity АОС

```

package com.setup.kineticar;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

import com.google.android.material.navigation.NavigationView;

import androidx.drawerlayout.widget.DrawerLayout;

import androidx.appcompat.widget.Toolbar;

import android.view.Menu;

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration mAppBarConfiguration;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        /*FloatingActionButton fab = findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
            }
        });*/
        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        NavigationView navigationView = findViewById(R.id.nav_view);
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        mAppBarConfiguration = new AppBarConfiguration.Builder(
            R.id.nav_main_menu, R.id.nav_tasks_list,
            R.id.nav_settings, R.id.nav_directory_context,
R.id.nav_to_experiment)
            .setDrawerLayout(drawer)
            .build();
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController(this, navController,
mAppBarConfiguration);
        NavigationUI.setupWithNavController(navigationView, navController);
    }
}

```

```

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController(this,
            R.id.nav_host_fragment);
        return NavigationUI.navigateUp(navController, mAppBarConfiguration)
            || super.onSupportNavigateUp();
    }
}

```

Листинг Б.2 – Код класса Workspace АОС

```

package com.setup.kineticar;

import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.app.ActivityManager;
import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.Display;
import android.graphics.Point;
import android.widget.TextView;
import android.widget.Toast;

import com.google.ar.core.Anchor;
import com.google.ar.sceneform.AnchorNode;
import com.google.ar.sceneform.ux.ArFragment;
import com.setup.kineticar.Experiments.CurrentExperiment;
import com.setup.kineticar.Helpers.CameraPermissionHelper;
import com.setup.kineticar.Managers.CustomSettingsManager;
import com.setup.kineticar.Managers.SceneManager;
import com.setup.kineticar.Managers.TimeManager.TimeManager;
import com.setup.kineticar.R;

public class Workspace extends AppCompatActivity {

    private static final String TAG = "No AR";
    private static final double MIN_OPENGL_VERSION = 3.0;

```

```

private TextView _textView;
private ArFragment _arFragment;
private CurrentExperiment _instruments;

@Override
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    if (!CameraPermissionHelper.hasCameraPermission(this)) {
        CameraPermissionHelper.requestCameraPermission(this);
        return;
    }

    if (!checkIsSupportedDeviceOrFinish(this)) {
        return;
    }

    CustomSettingsManager.SettingsInitialization();
    setContentView(R.layout.activity_work_space);

    Display display = getWindowManager().getDefaultDisplay();
    Point size = new Point();
    display.getRealSize(size);

    _arFragment = (ArFragment)
getSupportFragmentManager().findFragmentById(R.id.arFragment);
    _textView = findViewById(R.id.instrument_name);

    String currentInstrument = getIntent().getExtras().getString("instrument");

    _textView.setText(getString(R.string.current_instrument_en) + "\n" +
currentInstrument);

    CustomSettingsManager.getDisplaySettings().SetDisplaySize(size.x, size.y);
    SceneManager.setCamera(_arFragment);
    TimeManager.setTimerStopWatch();

    _instruments = new CurrentExperiment();
    _instruments.setInstrument(_arFragment.getTransformationSystem(),
currentInstrument, this);

    _arFragment.setOnTapArPlaneListener((hitResult, plane, motionEvent) -> {
        Anchor part1Anchor = hitResult.createAnchor();
        Anchor part2Anchor = hitResult.createAnchor();

        AnchorNode part1Node = new AnchorNode(part1Anchor);
        AnchorNode part2Node = new AnchorNode(part2Anchor);

        _instruments.getLaboratoryInstrument().loadAtScene(this, part1Node,
part2Node);

```

```

        _arFragment.getArSceneView().getScene().addChild(part1Node);
        _arFragment.getArSceneView().getScene().addChild(part2Node);
    });
}

public static boolean checkIsSupportedDeviceOrFinish(final Activity activity) {
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.N) {
        Log.e(TAG, "Sceneform requires Android N or later");
        Toast.makeText(activity, "Sceneform requires Android N or later",
            Toast.LENGTH_LONG).show();
        activity.finish();
        return false;
    }
    String openGlVersionString;
    openGlVersionString = ((ActivityManager)
        activity.getSystemService(Context.ACTIVITY_SERVICE))
        .getDeviceConfigurationInfo()
        .getGlEsVersion();
    if (Double.parseDouble(openGlVersionString) < MIN_OPENGL_VERSION) {
        Log.e(TAG, "Sceneform requires OpenGL ES 3.0 later");
        Toast.makeText(activity, "Sceneform requires OpenGL ES 3.0 or later",
            Toast.LENGTH_LONG)
            .show();
        activity.finish();
        return false;
    }
    return true;
}
}

```

Листинг Б.3 – Код класса MathPendulumExperoment AOC

```

package com.setup.kineticar.Experiments.PendulumsExperiments.MathPendulumExperiment;

import android.content.Context;

import com.google.ar.sceneform.Node;
import com.google.ar.sceneform.ux.TransformationSystem;
import com.setup.kineticar.Experiments.IExperiment;
import com.setup.kineticar.LaboratoryInstruments.Pendulums.MathPendulum.MathPendulum;

public class MathPendulumExperoment implements IExperiment {
    private MathPendulum _matchPendulum;

    public MathPendulumExperoment(TransformationSystem transformationSystem){
        _matchPendulum = new MathPendulum(transformationSystem);
    }

    @Override
    public void loadAtScene(Context scene, Node modelParent, Node movingPartParent) {
        _matchPendulum.loadAtScene(scene, modelParent, movingPartParent);
    }
}

```

Листинг Б.4 – Код класса MysubtopicscontentRecyclerViewAdapter АОС

```
package com.setup.kineticar.ui.DirectoryContent;

import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.setup.kineticar.ContentModel.Subtopic;
import com.setup.kineticar.MainActivity;
import com.setup.kineticar.R;
import com.setup.kineticar.dummy.DummyContent.DummyItem;

import java.util.List;

/**
 * {@link RecyclerView.Adapter} that can display a {@link DummyItem}.
 * TODO: Replace the implementation with code for your data type.
 */
public class MysubtopicscontentRecyclerViewAdapter extends
RecyclerView.Adapter<MysubtopicscontentRecyclerViewAdapter.ViewHolder> {

    private final List<Subtopic> mValues;
    boolean focus = true;
    Context context;

    public MysubtopicscontentRecyclerViewAdapter(List<Subtopic> items, Context
context) {
        mValues = items;
        this.context = context;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.main_menu_list_item, parent, false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(final ViewHolder holder, int position) {
        holder.textView.setText(mValues.get(position).get_name());
        holder.image.setImageResource(R.drawable.app_icon);
    }
}
```

```

        holder.subtopic = mValues.get(position);
    }

    @Override
    public int getItemCount() {
        return mValues.size();
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
        TextView textView;
        ImageView image;
        Button button;
        Subtopic subtopic;

        public ViewHolder(View view) {
            super(view);

            button = itemView.findViewById(R.id.list_button);
            textView = itemView.findViewById(R.id.list_text);
            image = itemView.findViewById(R.id.list_image);

            button.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    MainActivity activity = (MainActivity) context;
                    if (subtopic.get_subtopics() != null) {

                        SubtopicscontentFragmentDirections.SubtopicAction action =
SubtopicscontentFragmentDirections.subtopicAction(subtopic);
                        action.setSubtopic(subtopic);
                        NavController navController =
Navigation.findNavController(activity, R.id.nav_host_fragment);
                        navController.navigate(action);
                    }
                    else if(subtopic.get_paragraph() != null)
                    {
                        SubtopicscontentFragmentDirections.ParagraphAction action =
SubtopicscontentFragmentDirections.paragraphAction(subtopic);
                        action.setSubtopic(subtopic);
                        NavController navController =
Navigation.findNavController(activity, R.id.nav_host_fragment);
                        navController.navigate(action);
                    }
                    else
                    {
                        Toast.makeText(activity, "Содержимое недоступно",
Toast.LENGTH_LONG).show();
                    }
                }
            });
        }
    }
}

```



```

        Subtopic subtopic = null;
        if(getArguments() != null) {
            // The getPrivacyPolicyLink() method will be created automatically.
            subtopic =
SubtopicscontentFragmentArgs.fromBundle(getArguments()).getSubtopic();
        }

        View view = inflater.inflate(R.layout.fragment_subtopic, container, false);

        RecyclerView recyclerView = view.findViewById(R.id.list);
        recyclerView.setLayoutManager(new LinearLayoutManager(view.getContext()));
        recyclerView.setAdapter(new
MysubtopicscontentRecyclerViewAdapter(subtopic.get_subtopics(),view.getContext() ));
        return view;
    }
}

```

Листинг Б.6 – Код класса Subtopic АОС

```

package com.setup.kineticar.ContentModel;

import android.os.Parcel;
import android.os.Parcelable;

import java.util.List;

public class Subtopic implements Parcelable {
    private String _name;
    private String _paragraph;
    private List<Subtopic> _subtopics;

    protected Subtopic(Parcel in) {
        _name = in.readString();
        _paragraph = in.readString();
        _subtopics = in.createTypedArrayList(Subtopic.CREATOR);
    }

    public static final Creator<Subtopic> CREATOR = new Creator<Subtopic>() {
        @Override
        public Subtopic createFromParcel(Parcel in) {
            return new Subtopic(in);
        }

        @Override
        public Subtopic[] newArray(int size) {
            return new Subtopic[size];
        }
    };

    public String get_name() {
        return _name;
    }

    public List<Subtopic> get_subtopics() {
        return _subtopics;
    }
}

```

```

    }

    public String get_paragraph() {
        return _paragraph;
    }

    public Subtopic(String name, List<Subtopic> subtopics) {
        this._name = name;
        this._subtopics = subtopics;
    }

    public Subtopic(String name, String paragraph) {
        this._name = name;
        this._paragraph = paragraph;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(_name);
        dest.writeString(_paragraph);
        dest.writeTypedList(_subtopics);
    }
}

```

Листинг Б.7 – Код класса CurrentExperiment АОС

```

package com.setup.kineticar.Experiments;

import android.content.Context;

import com.google.ar.sceneform.ux.TransformationSystem;
import com.setup.kineticar.Experiments.AtwoodMachine.AtwoodMachineExperiment;
import com.setup.kineticar.Experiments.DensityDetermination.DensityDetermination;
import
com.setup.kineticar.Experiments.PendulumsExperiments.MathPendulumExperiment.MathPendu
lumExperoment;
import
com.setup.kineticar.Experiments.PendulumsExperiments.SpringPendulumExperiment.SpringP
endulumExperiment;
import com.setup.kineticar.LaboratoryInstruments.AtwoodMachine.AtwoodMachine;
import com.setup.kineticar.Experiments.BulletSpeedDetection.BulletSpeedDetection;
import com.setup.kineticar.LaboratoryInstruments.IInstrument;
import com.setup.kineticar.LaboratoryInstruments.Pendulums.MathPendulum.MathPendulum;
import
com.setup.kineticar.LaboratoryInstruments.Pendulums.SpringPendulum.SpringPendulum;

public class CurrentExperiment {

    private IExperiment _laboratoryInstrument;

    public IExperiment getLaboratoryInstrument() {
        return _laboratoryInstrument;
    }
}

```

```

    }

    public CurrentExperiment(){
    }

    public void setInstrument(TransformationSystem transformationSystem, String
currentInstrument, Context scene) {

        switch (currentInstrument){
            case "Math Pendulum":
                _laboratoryInstrument = new
MathPendulumExperoment(transformationSystem);
                break;
            case "Spring Pendulum":
                _laboratoryInstrument = new
SpringPendulumExperiment(transformationSystem);
                break;
            case "Atwood Machine":
                _laboratoryInstrument = new
AtwoodMachineExperiment(transformationSystem, scene);
                break;
            case "Bullet Speed Detection":
                _laboratoryInstrument = new
BulletSpeedDetection(transformationSystem, scene);
                break;
            case "Density Determination":
                _laboratoryInstrument = new
DensityDetermination(transformationSystem);
                break;
            default:
                _laboratoryInstrument = new
MathPendulumExperoment(transformationSystem);
                break;
        }

    }

}
}

```

Листинг Б.8 – Код класса IInstrument АОС

```

package com.setup.kineticar.LaboratoryInstruments;

import android.content.Context;

import com.google.ar.sceneform.AnchorNode;
import com.google.ar.sceneform.Node;
import com.google.ar.sceneform.math.Vector3;
import com.setup.kineticar.LaboratoryInstruments.DiskMachine.View.DiskMachineModels;

public interface IInstrument {
    public void loadAtScene(Context scene, Node modelParent, Node
movingPartParent);
}

```

Листинг Б.9 – Код класса InstrumentNode АОС

```

package com.setup.kineticar.LaboratoryInstruments;

import android.app.AlertDialog;
import android.content.Context;
import android.net.Uri;
import android.util.Log;

import com.google.ar.sceneform.Node;
import com.google.ar.sceneform.math.Vector3;
import com.google.ar.sceneform.rendering.ModelRenderable;
import com.setup.kineticar.CustomMaterial;
import
com.setup.kineticar.LaboratoryInstruments.AtwoodMachine.View.AtwoodMachineModels;

import java.util.concurrent.CompletableFuture;

public class InstrumentNode extends Node {
    protected String _view = AtwoodMachineModels.Load.getAsset();
    protected Vector3 _offsetFromParent;

    private CustomMaterial customMaterial;
    private ModelRenderable renderableModel;

    public void setView(String _view) {
        this._view = _view;
    }
    public String getView() {
        return _view;
    }
    }

    public void setOffsetFromParent(Vector3 offsetFromParent){
        this._offsetFromParent = offsetFromParent;
    }
    public Vector3 getOffsetFromParent(){
        return this._offsetFromParent;
    }
    }

    public CustomMaterial getCustomMaterial() {
        return customMaterial;
    }
    }

    public ModelRenderable getRenderableModel() {
        return renderableModel;
    }
    }

    public void setPosition(){
        setLocalPosition(_offsetFromParent);
    }
    }

    public void setTexturedModel(Context scene, String baseColorSource,
                                String metallicSource, String roughnessSource,
String normalSource){

        CompletableFuture<CustomMaterial> materialFuture =
CustomMaterial.build(scene,
        Uri.parse(baseColorSource),
        Uri.parse(metallicSource),
        Uri.parse(roughnessSource),

```

```

        Uri.parse(normalSource)
    );

    CompletableFuture<ModelRenderable> renderableFuture =
ModelRenderable.builder()
        .setSource(scene, Uri.parse(_view))
        .build();

    renderableFuture.thenAcceptBoth(materialFuture, (renderableResult,
materialResult)-> {
        Log.i("InstrumentNode", "init material and renderable");
        customMaterial = materialResult;
        renderableModel = renderableResult;
        customMaterial.setTexture();
        renderableModel.setMaterial(customMaterial.getValue());
    });
}

public void setModel() {
    this.setRenderable(getRenderableModel());
}

public void setModel(Context scene){

    ModelRenderable.builder()
        .setSource(scene, Uri.parse(_view))
        .build()
        .thenAccept((modelRenderable) -> this.setRenderable(modelRenderable))
        .exceptionally(throwable -> {
            AlertDialog.Builder builder = new AlertDialog.Builder(scene);
            builder.setMessage(throwable.getMessage())
                .show();
            return null;
        });
}
}

```

Листинг Б.10 – Код класса MathModel АОС

```

package com.setup.kineticar.LaboratoryInstruments.Pendulums.MathPendulum.MathModel;

import com.google.ar.sceneform.math.Vector3;
import
com.setup.kineticar.LaboratoryInstruments.Pendulums.MathPendulum.Thread.Thread;
import com.setup.kineticar.Mathematics.Geometry;

public class MathModel {
    protected Vector3 circleCenter = new Vector3();
    protected Vector3 neutralPosition = new Vector3();
    protected float angle = 0.0f;
    protected float time = 0.0f;
    protected float angleChange = 0.0f;
}

```

```

public void setCircleCenter(Vector3 circleCenter) {
    this.circleCenter = circleCenter;
}
public Vector3 getCircleCenter(){
    return this.circleCenter;
}

public void setNeutralPosition(Vector3 neutralPosition){
    this.neutralPosition = neutralPosition;
}
public Vector3 getNeutralPosition(){
    return this.neutralPosition;
}

public void setAngle(float angle){
    this.angle = angle;
}
public float getAngle(){
    return this.angle;
}

public Vector3 PerformStartPosition(Vector3 point, float threadLength){
    float R1 = Geometry.LengthOfLine(point.x, point.y,
        circleCenter.x, circleCenter.y);

    float sin = (point.x - circleCenter.x) / R1;
    float cos = -(float) Math.sqrt(1 - Math.pow(sin, 2));

    angle = (float)(Math.asin(sin));

    Vector3 newPosition = new Vector3(
        circleCenter.x + threadLength * sin,
        circleCenter.y + threadLength * cos,
        neutralPosition.z
    );
    return newPosition;
}

public Vector3 PerformModel(Vector3 position, Thread thread){
    return new Vector3();
}

}

```

Листинг Б.11 – Код класса MathModel1 АОС

```

package com.setup.kineticar.LaboratoryInstruments.Pendulums.MathPendulum.MathModel;

import com.google.ar.sceneform.math.Quaternion;
import com.google.ar.sceneform.math.Vector3;
import
com.setup.kineticar.LaboratoryInstruments.Pendulums.MathPendulum.Thread.Thread;
import com.setup.kineticar.Managers.TimeManager.TimeManager;
import com.setup.kineticar.Physics.EnvironmentParameters;

public class MathModel1 extends MathModel {

    @Override
    public Vector3 PerformModel(Vector3 position, Thread thread) {
        Vector3 newCoordinates = position;
        float newTime;
        newTime = TimeManager.getTimeSec();

        if(newTime != time) {
            time = newTime;

            angleChange -= time * EnvironmentParameters.g.getValue() /
thread.getLengthM() * Math.sin(angle)
                + time * EnvironmentParameters.p.getValue() * angleChange;

            angle += time * angleChange;

            float cos = (float) Math.cos(angle);
            float sin = (float) Math.sin(angle);

            float newDeltaX = thread.getLengthM() * sin;
            float newDeltaY = thread.getLengthM() * cos;

            float newX = neutralPosition.x + newDeltaX;
            float newY = circleCenter.y - newDeltaY;

            newCoordinates = new Vector3(newX, newY, position.z);

            //thread.setLocalRotation(Quaternion.eulerAngles(new Vector3(0.0f, 0.0f,
(float) Math.toDegrees(angle))));

            TimeManager.RestartStopWatch();
        }

        return newCoordinates;
    }
}

```

Листинг Б.12 – Код класса MathModel2 АОС

```

package com.setup.kineticar.LaboratoryInstruments.Pendulums.MathPendulum.MathModel;

import com.google.ar.sceneform.math.Quaternion;
import com.google.ar.sceneform.math.Vector3;
import com.setup.kineticar.LaboratoryInstruments.Pendulums.MathPendulum.Thread.Thread;
import com.setup.kineticar.Managers.TimeManager.TimeManager;
import com.setup.kineticar.Physics.EnvironmentParameters;

import java.time.temporal.ValueRange;

public class MathModel2 extends MathModel {
    private float newAngle;

    @Override
    public float getAngle(){
        return newAngle;
    }

    @Override
    public Vector3 PerformStartPosition(Vector3 point, float threadLength) {
        Vector3 newPosition = super.PerformStartPosition(point, threadLength);
        newAngle = angle;
        return newPosition;
    }

    @Override
    public Vector3 PerformModel(Vector3 position, Thread thread) {
        float newTime;
        Vector3 newCoordinates = position;
        newTime = TimeManager.getTimeSec();
        if (time != newTime) {

            time = newTime;
            newAngle = (float) (angle * Math.sin(Math.PI / 2 + Math.sqrt(EnvironmentParameters.g.getValue() /
            thread.getLengthM()) * time));

            float cos = (float) Math.cos(newAngle);
            float sin = (float) Math.sin(newAngle);

            float newDeltaX = thread.getLengthM()*sin;
            float newDeltaY = thread.getLengthM() * cos;

            float newX = neutralPosition.x + newDeltaX;
            float newY = circleCenter.y - newDeltaY;

            newCoordinates = new Vector3(newX, newY, position.z);

            //thread.setLocalRotation(Quaternion.eulerAngles(new Vector3(0.0f, 0.0f, (float)
            Math.toDegrees(newAngle))));
        }

        return newCoordinates;
    }
}

```

Листинг Б.13 – Код класса Axis АОС

```
package com.setup.kineticar.LaboratoryInstruments.Pendulums.MathPendulum.Pedestal;

import com.setup.kineticar.LaboratoryInstruments.InstrumentNode;
import com.setup.kineticar.LaboratoryInstruments.Pendulums.MathPendulum.View.MathPendulumsModels;

public class Axis extends InstrumentNode {
    public Axis(){
        super();
        _view = MathPendulumsModels.PedestalAxis.getAsset();
    }
}
```

Приложение В

СВЯЗИ МЕЖДУ ЭЛЕМЕНТАМИ ОНТОЛОГИЧЕСКОЙ МОДЕЛИ

(рекомендуемое)

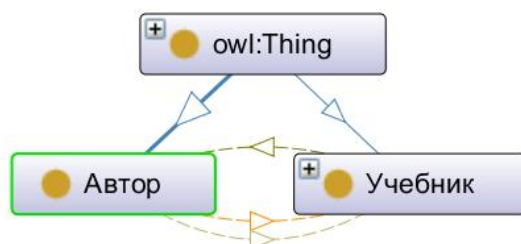


Рисунок В.1 – Класс «Автор» и его связи

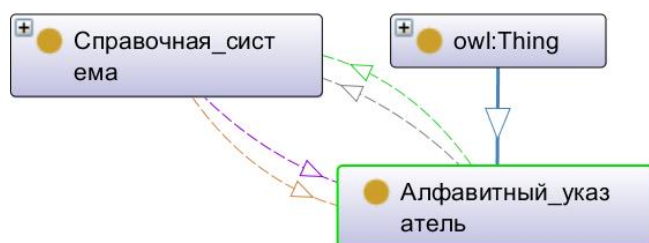


Рисунок В.2 – Класс «Алфавитный указатель» и его связи

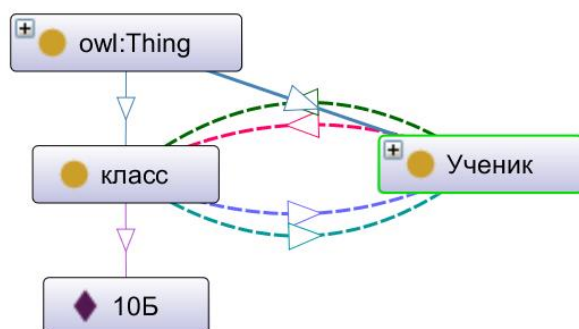


Рисунок В.3 – Класс «Ученик» и его связи

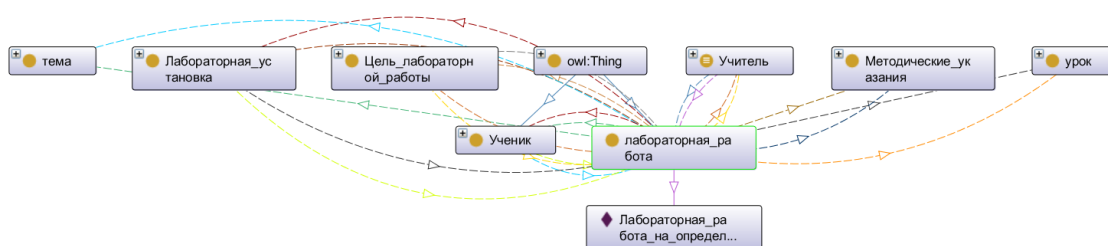


Рисунок В.4 – Класс «Лабораторная работа» и его связи

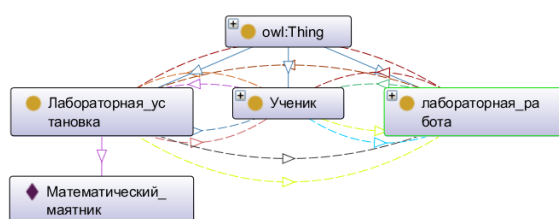


Рисунок В.5 – Класс «Лабораторная установка» и его связи

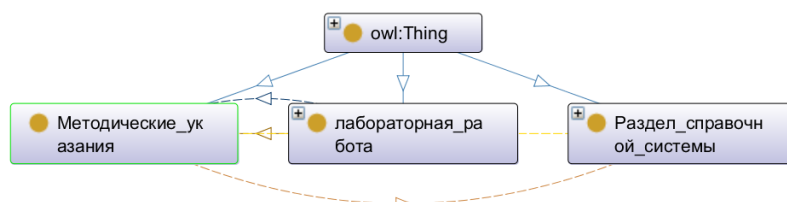


Рисунок В.6 – Класс «Методические указания» и его связи

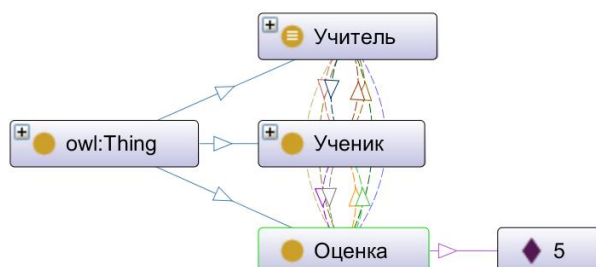


Рисунок В.7 – Класс «Оценка» и его связи

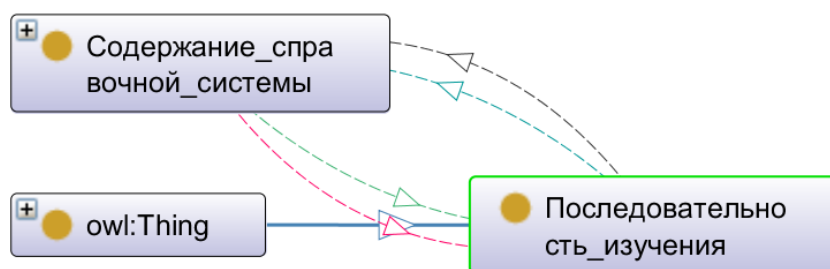


Рисунок В.8 – Класс «Последовательность изучения» и его связи

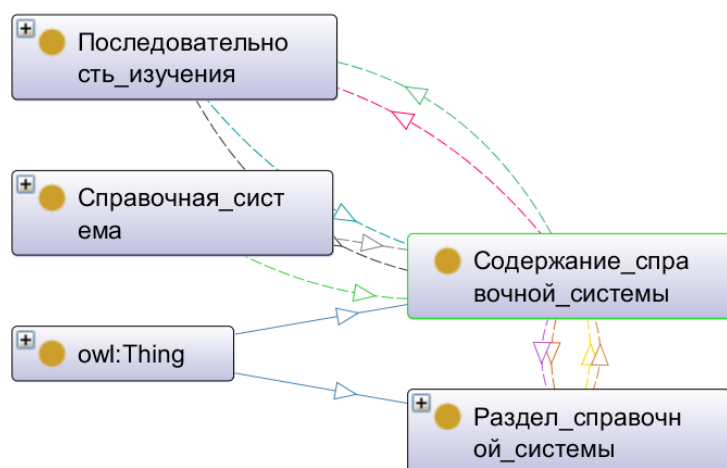


Рисунок В.9 – Класс «Содержание справочной системы» и его связи

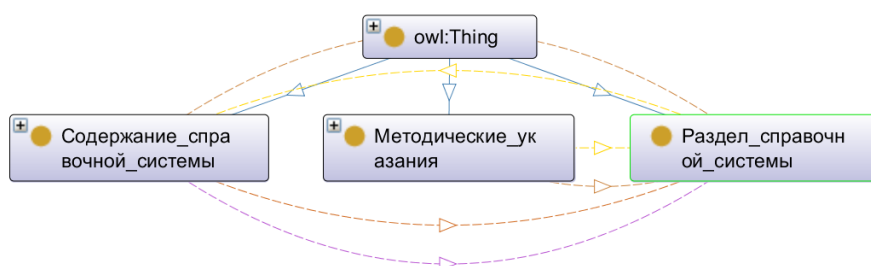


Рисунок В.10 – Класс «Раздел справочной системы» и его связи

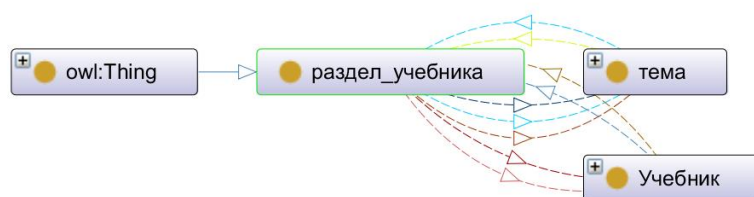


Рисунок В.11 – Класс «Раздел учебника» и его связи

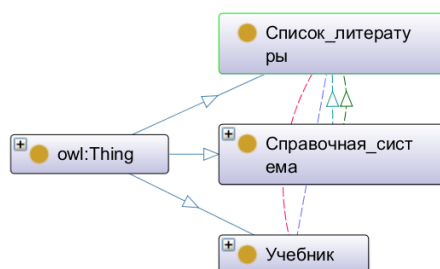


Рисунок В.12 – Класс «Список литературы» и его связи

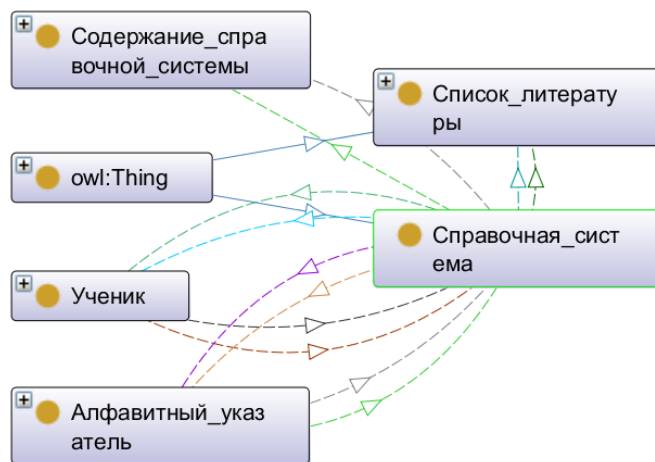


Рисунок В.13 – Класс «Справочная система» и его связи

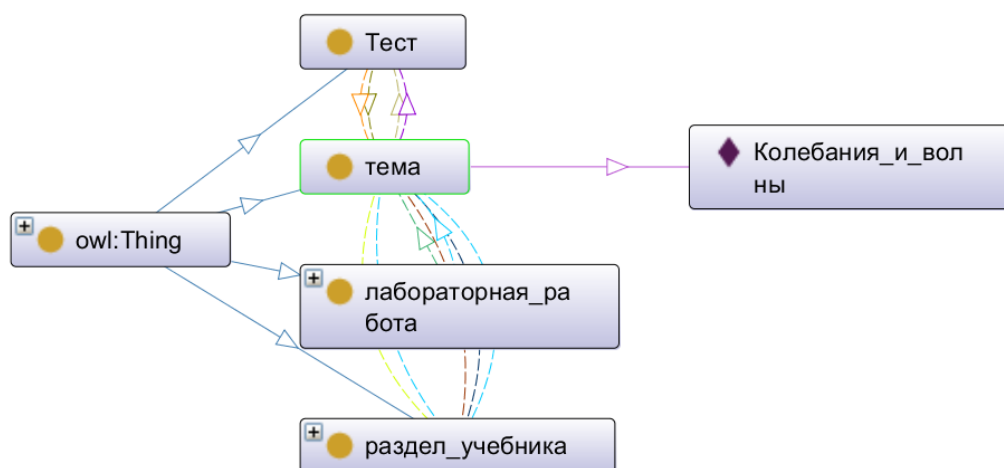


Рисунок В.14 – Класс «Тема» и его связи



Рисунок В.15 – Класс «Урок» и его связи

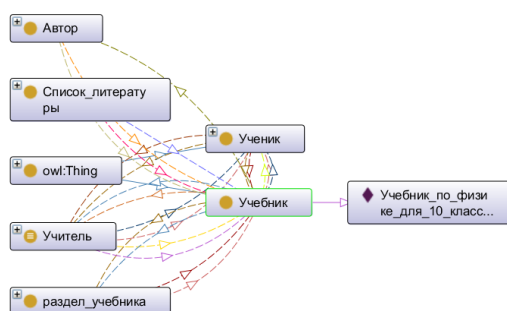


Рисунок В.16 – Класс «Учебник» и его связи

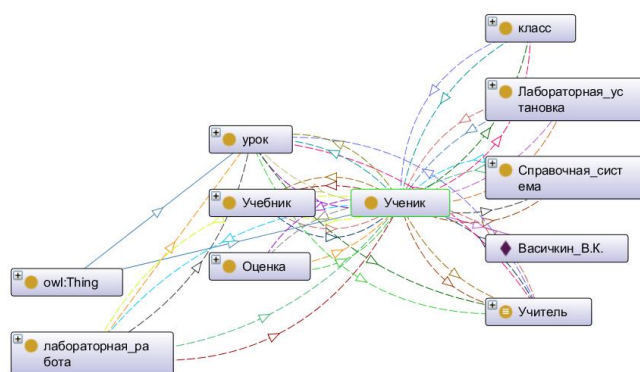


Рисунок В.17 – Класс «Ученик» и его связи

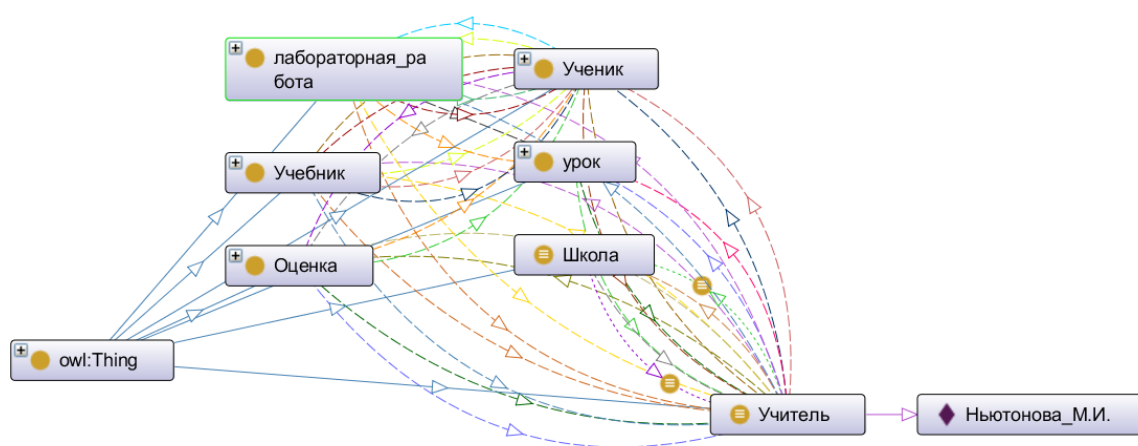


Рисунок В.18 – Класс «Учитель» и его связи

Приложение Г

СКРИНШОТЫ ПРОТОТИПА СПРАВОЧНОЙ ЧАСТИ АОС (рекомендуемое)

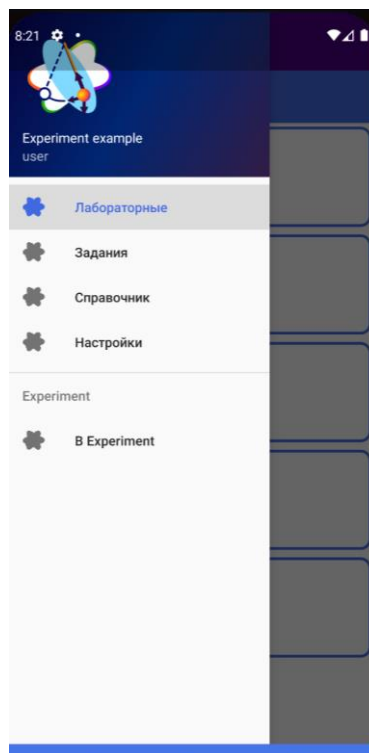


Рисунок Г.1 – Главное меню



Рисунок Г.2 – Окно со списком лабораторных работ

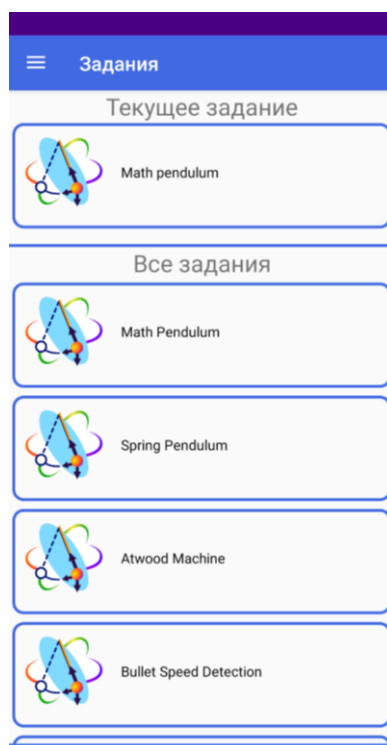


Рисунок Г.3 – Окно со списком заданий

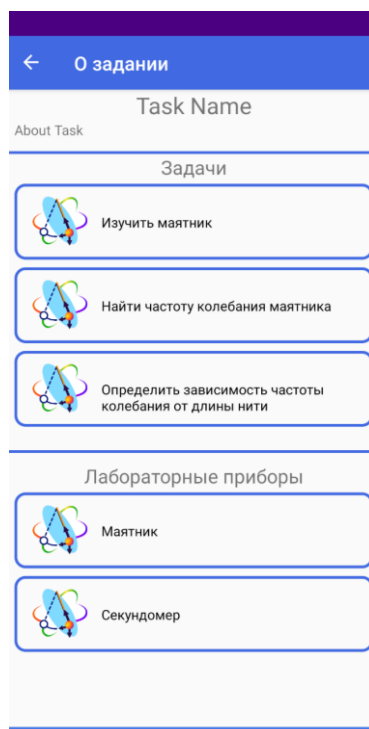


Рисунок Г.4 – Окно со информацией о задании

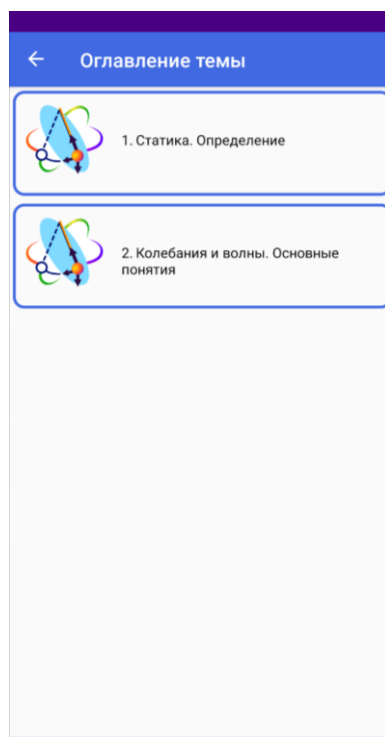


Рисунок Г.5 – Окно оглавления справочника

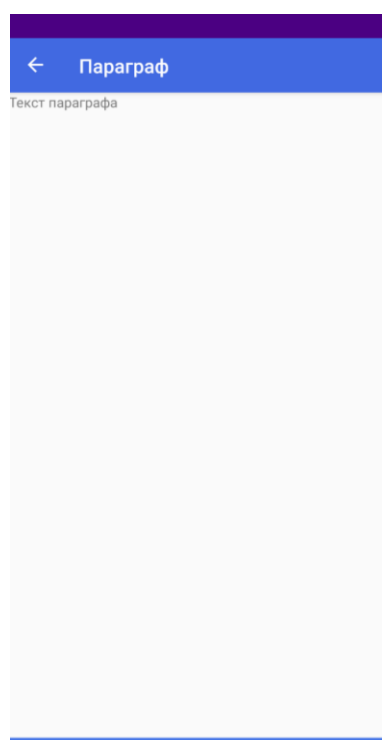


Рисунок Г.6 – Окно параграфа справочника

Приложение Д

**ДИПЛОМ ЛАУРЕАТА I СТЕПЕНИ В СЕКЦИИ «ТЕХНИЧЕСКИЕ
НАУКИ» XXIII МЕЖДУНАРОДНОГО НАУЧНО-
ИССЛЕДОВАТЕЛЬСКОГО КОНКУРСА «ЛУЧШАЯ НАУЧНАЯ
СТАТЬЯ 2019»**

(рекомендуемое)



DIPLOMA

I степени
в секции «Технические науки»
XXIII Международного научно-исследовательского конкурса
«ЛУЧШАЯ НАУЧНАЯ СТАТЬЯ 2019»

НАГРАЖДАЕТСЯ
Акчурина Варвара Дмитриевна
студент
Пензенский государственный университет

автор научной работы:
«РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ «ФИЗИЧЕСКАЯ
ЛАБОРАТОРИЯ» С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ
ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ»
научный руководитель: к.т.н. доцент, доцент кафедры «МО и ПЭВМ»,
Князев Виктор Николаевич

28 февраля 2019 г.,
г. Пенза, Российская Федерация
Директор МЦНС
«Наука и просвещение»
к.э.н. Гуляев Г.Ю.



НАУКА и ПРОСВЕЩЕНИЕ
МЕЖДУНАРОДНЫЙ ЦЕНТР НАУЧНОГО СОТРУДНИЧЕСТВА

НАУЧНАЯ ЭЛЕКТРОННАЯ
БИБЛИОТЕКА
LIBRARY.RU



VISA



Приложение Е

**ДИПЛОМ ЛАУРЕАТА I СТЕПЕНИ XII МЕЖДУНАРОДНОГО
МОЛОДЕЖНОГО КОНКУРСА НАУЧНЫХ РАБОТ «МОЛОДЕЖЬ В
НАУКЕ: НОВЫЕ АРГУМЕНТЫ»**

(рекомендуемое)



XII МЕЖДУНАРОДНЫЙ МОЛОДЕЖНЫЙ КОНКУРС НАУЧНЫХ РАБОТ
«МОЛОДЕЖЬ В НАУКЕ: НОВЫЕ АРГУМЕНТЫ»
 РОССИЙСКАЯ ФЕДЕРАЦИЯ, Г. ЛИПЕЦК
 1 ноября 2019 г.

№ 69/11–19

22 ноября 2019

ДИПЛОМ

ЛАУРЕАТА I СТЕПЕНИ

Настоящий диплом подтверждает, что

Аксурин Варвара Дмитриевна

удостоен(а) почетного звания Лауреата I степени
 XII Международного молодежного конкурса научных работ
 «Молодежь в науке: Новые аргументы» от 1 ноября 2019 г.

ТЕМА НАУЧНОЙ РАБОТЫ:

«Разработка автоматизированной обучающей системы с
 использованием технологии дополненной реальности»

НАУЧНЫЙ РУКОВОДИТЕЛЬ:

канд. техн. наук, доцент Князев Виктор Николаевич

Председатель Конкурсной Комиссии
 О.В. Федоров



Ученый секретарь
 А.В. Горбунко



Россия, Липецк — 2019 г.

Приложение Ж

**ГРАМОТА ЗА ПОБЕДУ В НОМИНАЦИИ «ЗА ЛУЧШЕЕ
ОФОРМЛЕНИЕ» XII МЕЖДУНАРОДНОГО МОЛОДЕЖНОГО
КОНКУРСА НАУЧНЫХ РАБОТ «МОЛОДЕЖЬ В НАУКЕ: НОВЫЕ
АРГУМЕНТЫ»**

(рекомендуемое)



XII МЕЖДУНАРОДНЫЙ МОЛОДЕЖНЫЙ КОНКУРС НАУЧНЫХ РАБОТ
«МОЛОДЕЖЬ В НАУКЕ: НОВЫЕ АРГУМЕНТЫ»
 РОССИЙСКАЯ ФЕДЕРАЦИЯ, г. ЛИПЕЦК
 1 ноября 2019 г.

№ 01/11-19

ГРАМОТА

22 ноября 2019

За лучшее оформление

Награждается

Акиурин Варвара Дмитриевна

участник XII Международного молодежного конкурса научных работ
 «Молодежь в науке: Новые аргументы» от 1 ноября 2019 г.

ТЕМА НАУЧНОЙ РАБОТЫ:

«Разработка автоматизированной обучающей системы с
 использованием технологии дополненной реальности»

НАУЧНЫЙ РУКОВОДИТЕЛЬ:

канд. техн. наук, доцент Князева Виктор Николаевич

Председатель Конкурсной Комиссии
 О.В. Федоров



Ученый секретарь
 А.В. Горбенко

ДЛЯ ДОКУМЕНТОВ

«АРГУМЕНТ» «ARGUMENT»
 Россия, Липецк | Lipetsk, Russia
 +7 (4742) 39-79-73



Россия, Липецк — 2019 г.

Приложение И

**ГРАМОТА ЗА ПОБЕДУ В НОМИНАЦИИ «ЗА ЛУЧШИЙ
ГРАФИЧЕСКИЙ МАТЕРИАЛ» XII МЕЖДУНАРОДНОГО
МОЛОДЕЖНОГО КОНКУРСА НАУЧНЫХ РАБОТ «МОЛОДЕЖЬ В
НАУКЕ: НОВЫЕ АРГУМЕНТЫ»**

(рекомендуемое)



ХII МЕЖДУНАРОДНЫЙ МОЛОДЕЖНЫЙ КОНКУРС НАУЧНЫХ РАБОТ
«МОЛОДЕЖЬ В НАУКЕ: НОВЫЕ АРГУМЕНТЫ»
 РОССИЙСКАЯ ФЕДЕРАЦИЯ, г. ЛИПЕЦК
 1 ноября 2019 г.

№ 10/11-19

ГРАМОТА

22 ноября 2019

За лучший графический материал

Награждается

Актурина Варвара Дмитриевна

участник XII Международного молодежного конкурса научных работ
 «Молодежь в науке: Новые аргументы» от 1 ноября 2019 г.

ТЕМА НАУЧНОЙ РАБОТЫ:

«Разработка автоматизированной обучающей системы с
 использованием технологии дополненной реальности»

НАУЧНЫЙ РУКОВОДИТЕЛЬ:

канд. техн. наук, доцент Князь Виктор Николаевич

Председатель Конкурсной Комиссии
 О.В. Федоров



Ученый секретарь
 А.В. Горбунко

ДЛЯ ДОКУМЕНТОВ
 Печать конкурсного оргкомитета
 «АРГУМЕНТ» «ARGUMENT»
 Россия, Липецк | Lipetsk, Russia
 +7 (4742) 39-79-73



Россия, Липецк – 2019 г.

Приложение К

**СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ПРОГРАММЫ
ФОРМИРОВАНИЯ РАСШИРЕННОЙ МОДЕЛИ ОСВОЕНИЯ
УЧЕБНОГО МАТЕРИАЛА**

(рекомендуемое)

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ
№ 2021612391

**Программа формирования расширенной модели
освоения учебного материала**

Правообладатель: **Акчурина Варвара Дмитриевна (RU)**

Авторы: **Акчурина Варвара Дмитриевна (RU), Князев
Виктор Николаевич (RU)**

Заявка № **2020664015**

Дата поступления **03 ноября 2020 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **16 февраля 2021 г.**



Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев

Приложение Л

АКТ О ВНЕДРЕНИИ

(рекомендуемое)

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ ГОРОДА ПЕНЗЫ

**Муниципальное бюджетное общеобразовательное учреждение
средняя общеобразовательная школа
с углубленным изучением информатики № 68 г. Пензы
(МБОУ СОШ с углубленным изучением информатики № 68)**

пр. Строителей, д. 140, г. Пенза, 440064 (корпус №1)

пр. Строителей, д. 128, г. Пенза, 440071 (корпус №2)

тел. (8412) 41-14-20, 41-49-89, E-mail: school68@guoedu.ru

ОКПО 23994479, ОГРН 1025801219223

ИНН/КПП 5835005689/583501001

АКТ

о внедрении результатов диссертационной работы Акчуриной В.Д.

Настоящим Актом удостоверяется, что результаты диссертационного исследования Акчуриной В.Д. на тему «Исследование и разработка автоматизированных обучающих систем с использованием технологии дополненной реальности» проходят опытную эксплуатацию в МБОУ СОШ с углублённым изучением информатики №68 г.Пензы. Разработанная Программа формирования расширенной модели освоения учебного материала используются для улучшения структуры материала учебной программы по курсу информатики. С учетом предложенных в диссертационной работе рекомендаций ведется дальнейшее развитие программы.

Директор МБОУ СОШ
с углублённым изучением информатики
№68 г.Пензы



A handwritten signature in blue ink, appearing to be 'Х.Г. Курмаев'.

Х.Г.Курмаев