МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО "ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника»

"___" _____ 20 г.

Заведующий кафедрой

_____ М. А. Митрохин

ОТЧЕТ ПО ПРАКТИКЕ ПО ПОЛУЧЕНИЮ ПРОФЕССИОНАЛЬНЫХ УМЕНИЙ И ОПЫТА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ (В ТОМ ЧИСЛЕ ПЕДАГОГИЧЕСКАЯ ПРАКТИКА)

(2019/2020 учебный год)

Патунин Дмитрий Васильевич

Направление подготовки 09.04.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

Форма обучения — очная Срок обучения в соответствии с ФГОС — 2 года

Год обучения **2** семестр **4**

Период прохождения практики с 13.04.2020 по 26.04.2020

Кафедра «Вычислительная техника»

Заведующий кафедрой д. т. н., профессор, Митрохин М. А.

(должность, ученая степень, ученое звание, Ф.И.О.)

Научный руководитель доцент, к. т. н., Коннов Н. Н.

(должность, ученая степень, ученое звание)

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО "ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника»

"___" _____ 20 г.

Заведующий кафедрой

_____ М. А. Митрохин

ИНДИВИДУАЛЬНЫЙ ПЛАН
ПРОХОЖДЕНИЯ ПРАКТИКИ ПО ПОЛУЧЕНИЮ ПРОФЕССИОНАЛЬНЫХ УМЕНИЙ И ОПЫТА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ (В ТОМ ЧИСЛЕ ПЕДАГОГИЧЕСКАЯ ПРАКТИКА)
(2019/2020 учебный год)

Патунин Дмитрий Васильевич

Направление подготовки 09.04.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

Форма обучения — очная Срок обучения в соответствии с ФГОС — 2 года

Год обучения 2 семестр 4

Период прохождения практики:

с "13" апреля 2020 г. по "26" апреля 2020 г.

Кафедра «Вычислительная техника»

Заведующий кафедрой д. т. н., профессор, Митрохин М. А.

(должность, ученая степень, ученое звание, Ф.И.О.)

Научный руководитель доцент, к. т. н., Коннов Н. Н.

(должность, ученая степень, ученое звание)

№ п/п	Планируемые формы работы во время практики	Количество часов	Календарные сроки проведения планируемой работы	Подпись руководителя практики
1	Разработка индивидуального плана практики и его согласование с руководителем	2	13.04.2020 г.	
2	Консультирование (дистанционное) с руководителем о нормативно методической базой организации учебного процесса в вузе	1	13.04.2020 г.	
3	Самостоятельное изучение рекомендованной нормативно методической базой организации учебного процесса в вузе	12	13.04.2020 г. – 16.04.2020 г.	
4	Консультирование (дистанционное) с руководителем об основной образовательной программе кафедры ВТ	1	13.04.2020 г.	
5	Самостоятельное изучение рекомендованной образовательной программы кафедры ВТ	12	13.04.2020 г. – 16.04.2020 г.	
6	Консультирование (дистанционное) с руководителем плана ведения практики «Научно-исполнительская практика»	3	16.04.2020 г.	
7	Изучение учебно-методического материала практики	2	17.04.2020 г.	
8	Разработка заданий для группы по практике	8	13.04.2020 г. – 14.04.2020 г.	
9	Подготовка к проведению консультаций (дистанционно) по практике	10	13.04.2020.г. – 14.04.2020 г.	
10	Проведение консультаций (дистанционно) по практике «Научно-исполнительская практика»	8	13.04.2020. – 17.04.2020 г.	
11	Организация групповой работы. Ознакомление с программой практики, порядком прохождения, защиты отчета по практике и требованиями к оформлению отчета	3	17.04.2020 г.	
12	Систематизация учебного материала Распределение заданий для группы по практике.	8	16.04.2020 г. – 17.04.2020 г.	
13	Анализ, обсуждение и выполнение студентами индивидуального задания	12	13.04.2020 г. – 24.04.2020 г.	
14	Консультирование (дистанционное) с промежуточными версиями выполнения индивидуального задания	12	16.04.2020 г. – 24.04.2020 г.	
15	Проверка результатов у подгрупп студентов	10	18.04.2020 г. 21.04.2020 г. 24.04.2020 г.	
16	Анализ результатов проведения учебных занятий с руководителем	2	25.04.2020 г.	
17	Подготовка отчета о практике	2	25.04.2020 г.	
	Общий объем часов	108	13.04.2020 г. – 26.04.2020 г.	

Магистрант Патунин Д. В. _____ "___" _____ 2020 г.

Руководитель практики Коннов Н. Н. _____ "___" _____ 2020 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО "ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЧЕТ
О ПРОХОЖДЕНИИ ПРАКТИКИ ПО ПОЛУЧЕНИЮ ПРОФЕССИОНАЛЬНЫХ УМЕНИЙ И ОПЫТА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ (В ТОМ ЧИСЛЕ ПЕДАГОГИЧЕСКАЯ ПРАКТИКА)

(2019/2020 учебный год)

Патунин Дмитрий Васильевич

Направление подготовки 09.04.01 Информатика и вычислительная техника

Наименование профиля подготовки Вычислительные машины, комплексы, системы и сети

Форма обучения – очная Срок обучения в соответствии с ФГОС – 2 года

Год обучения **2** семестр **4**

Период прохождения практики:

с "13" апреля 2020 г. по "26" апреля 2020 г.

Кафедра «Вычислительная техника»

Основные итоги практики:

Патунин Д. В. вел у группы 17ВВ1 (2 курс) практику (дистанционно) «Научно-исполнительская практика». На первоначальном этапе были изучены и проанализированы учебно-методические материалы по учебной программе. А так же разработан и согласован с руководителем практики список вариантов заданий на практику. Произвел организацию групповой работы. Ознакомил с программой прохождения, защиты отчета по практике и требованиями к оформлению отчетов. Систематизировал учебно-методический материал для студентов. Выдал групповые задания. Консультировал в течение всей практики по вопросам, возникающим в процессе выполнения практики, с контролем промежуточных версий.

Магистрант Патунин Д. В. _____ "___" _____ 2020 г.

Руководитель практики Коннов Н. Н. _____ "___" _____ 2020 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО "ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЗЫВ

О ПРОХОЖДЕНИИ ПРАКТИКИ ПО ПОЛУЧЕНИЮ ПРОФЕССИОНАЛЬНЫХ УМЕНИЙ И ОПЫТА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ (В ТОМ ЧИСЛЕ ПЕДАГОГИЧЕСКАЯ ПРАКТИКА)

(2019/2020 учебный год)

Патунин Дмитрий Васильевич

(Ф.И.О. магистранта)

Направление подготовки 09.04.01 Информатика и вычислительная техника

Код, название

Наименование профиля подготовки Вычислительные машины, комплексы, системы и сети

Форма обучения — очная Срок обучения в соответствии с ФГОС — 2 года

Год обучения 2 семестр 4

Период прохождения практики:

с "13" апреля 2020 г. по "26" апреля 2020 г.

Кафедра «Вычислительная техника»

В процессе прохождения практики Патунин Д. В. решал следующие задачи: закрепление и углубление теоретических знаний в сфере информатики и вычислительной техники, а также приобретение практических навыков и компетенций в сфере научной, педагогической деятельности по профилю подготовки.

За период прохождения практики магистрантом были проведены подготовительные работы для организации и проведения практики (дистанционно), консультации (дистанционно) с подробным анализом и обсуждением (дистанционно) порядка проведения и сдачи (дистанционно) практики, а также организация групповой работы, выдача групповых заданий на практику и прием отчетов по ним (дистанционно). Во время практики показал себя ответственным, добросовестным руководителем, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике, программированию, знаниями о современных технологиях образовательного процесса. Патунин Д. В. свободно ориентировался во всех аспектах методики управления групповой работой во время практики.

Практика Патунина Д. В. заслуживает оценки «*отлично*».

Руководитель практики Коннов Н. Н. "___" _____ 2020 г.

Содержание

Цель работы 7
Ход выполнения работы 7
Общие сведения 8
Объектно-ориентированное программирование (ООП) 8
Класс 9
Указатели на объекты классов 11
Лабораторное задание 12
Варианты заданий 12
Требования к программе 13
Требования к классу 13
Требования к оформлению таблиц (ГОСТ 7.32-2001) 13
Требования к оформлению схем 15
Групповая работа 16
Подгруппа 1 17
Подгруппа 2 17
Подгруппа 3 17
Студент 1 17
Студент 2 18
Студент 3 18
Заключение 19
Список использованных источников 20
Приложение А. Пример оформления титульных листов 21

Цель работы

Целью учебной практики является формирование компетенций, необходимых студенту при осуществлении всех видов профессиональной деятельности, предусмотренных ФГОС ВО по направлению подготовки «Информатика и вычислительная техника», а также способствующих социальной мобильности и конкурентоспособности студента на рынке труда.

Ведение научно-исполнительской практики у студентов бакалавриата 2-ого курса происходило дистанционно. Дистанционное обучение (ДО) — взаимодействие учителя и учащихся между собой на расстоянии, отражающее все присущие учебному процессу компоненты (цели, содержание, методы, организационные формы, средства обучения) и реализуемое специфичными средствами Интернет-технологий или другими средствами, предусматривающими интерактивность.

Для достижения цели практики решались следующие задачи:

- закрепление и углубление теоретических знаний в сфере информатики и вычислительной техники;
- приобретение практических навыков и компетенций в сфере научной, педагогической и практической деятельности по профилю подготовки;
- создание учебно-методических материалов, в том числе электронных версий, по программе подготовки бакалавров.

Теоретической и методологической основой выполненной работы являются существующие материалы, используемые при организации научно-исполнительской практики бакалавров 2 курса.

При прохождении учебной практики используются подходы группового разделения сфер исследования существующих источников, включая научную, профессионально-техническую, и учебно-методическую литературу, анализа полученных результатов и консолидации результатов индивидуальных исследований в обобщенный учебно-методический материал.

При выполнении данной работы учитывается индивидуальный опыт, полученный в ходе выполнения группового задания и анализа учебно-методического материала в течение практики.

Ход выполнения работы

В ходе прохождения учебной практики был выполнен анализ учебной программы бакалавров 2 курса, и на ее основе были разработаны учебно-методические указания для прохождения учебной практики.

Учебно-методические материалы содержат 20 вариантов заданий на выполнение декомпозиции определенной предметной области для выделения в ней основных аспектов с целью дальнейшего представления в виде классов на языке C++. Для заданий определены общие требования к реализации программ и содержимому отчетов о результатах их выполнения.

Группа студентов бакалавриата 2 курса была разделена на подгруппы по 2 человека. Каждая бригада самостоятельно выбрала один из 20 вариантов заданий. Работа в бригадах осуществлялась по следующему сценарию:

1. студенты производят анализ предметной области заданной вариантом;
2. описывают предметную область в виде диаграммы классов;
3. студент реализует класс на языке C++
4. производят тестирование и отладку написанных классов.
5. объединение и использование реализованных классов в одной программе.

Для выполнения предложенных заданий учебной практики необходимы навыки, полученные в ходе изучения курса «Программирование», а также базовые знания по алгоритмизации. Для реализации программ допустимо использовать любые доступные языки программирования, за исключением случаев использования готовых или стандартных библиотек. Предпочтительным является использование языка C++.

При проверке качества реализации программы необходимо учитывать корректность декомпозиции предметной области, а также реакцию программы на «граничные» наборы данных, при которых возможно выявление ошибок реализации основного алгоритма или кода ввода и вывода данных.

Содержимое разработанных учебно-методических указаний представлено ниже.

Общие сведения

Объектно-ориентированное программирование (ООП)

Объектно-ориентированное программирование (ООП) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

Идеологически ООП — подход к программированию как к моделированию информационных объектов, решающий на новом уровне основную задачу структурного программирования: структурирование информации с точки зрения управляемости, что существенно улучшает управляемость самим процессом моделирования, что, в свою очередь, особенно важно при реализации крупных проектов.

Управляемость для иерархических систем предполагает минимизацию избыточности данных (аналогичную нормализации) и их целостность, поэтому созданное удобно управляемым — будет и удобно пониматься. Таким образом, через тактическую задачу управляемости решается стратегическая задача — транслировать понимание задачи программистом в наиболее удобную для дальнейшего использования форму.

Основные принципы структурирования в случае ООП связаны с различными аспектами базового понимания предметной задачи, которое требуется для оптимального управления соответствующей моделью:

- абстрагирование для выделения в моделируемом предмете важного для решения конкретной задачи по предмету, в конечном счёте — контекстное понимание предмета, формализуемое в виде класса;
- инкапсуляция для быстрой и безопасной организации собственно иерархической управляемости: чтобы было достаточно простой команды «что делать», без одновременного уточнения как именно делать, так как это уже другой уровень управления;
- наследование для быстрой и безопасной организации родственных понятий: чтобы было достаточно на каждом иерархическом шаге учитывать только изменения, не дублируя всё остальное, учтённое на предыдущих шагах;
- полиморфизм для определения точки, в которой единое управление лучше распараллелить или наоборот — собрать воедино.

То есть фактически речь идёт о прогрессирующей организации информации согласно первичным семантическим критериям: «важное/неважное», «ключевое/подробности», «родительское/дочернее», «единое/множественное». Прогрессирование, в частности, на последнем этапе даёт возможность перехода на следующий уровень детализации, что замыкает общий процесс.

Класс

Класс описывает абстрактное поведение. Объектные типы строятся на основе класса посредством добавления различных приватных полей и методов. **Объект** (то есть значение объектного типа, традиционно называемое «экземпляром класса») порождается **конструктором** на основе начальных параметров.

Класс — это определяемый пользователем тип. Класс определяется с использованием ключевого слова **class**.

```
class A {
public:
     A(void);  // Конструктор
     A(int);   // Конструктор
     ~A();     // Деструктор

     void func1(void);
     void func2(void);

protected:
     void func3(void);
     void func4(void);

private:
     int value;
```

```
};
```

Объект класса — это экземпляр класса (элемент типа класс или переменная типа класс). При создании объекта класса в памяти выделяется область, необходимая для размещения данных класса.

В языке C++ существует 3 спецификатора доступа: **public**, **protected**, **private**. Все спецификаторы являются ключевыми словами языка.

public — публичный, общедоступный, открытый. Доступ к члену класса, объявленного как public имеет любой пользователь класса.

protected — защищенный. Доступ к члену класса, объявленного как protected имеют:

- в данном классе;
- в классах, производных от данного класса;
- в классах и функциях, объявленных как дружественные данному классу (**friend**).

private — приватный, частный. Доступ к члену класса, объявленного как private имеют:

- в данном классе;
- в классах и функциях, объявленных как дружественные данному классу (**friend**).

После объявления спецификатора доступа все нижеследующие члены класса относятся к указанному спецификатору.

Конструктор — это функция инициализации объекта данного класса. Имя конструктора совпадает с именем класса. Конструктор не имеет возвращаемого значения. Допускается использование нескольких конструкторов для одного класса. В этом случае каждый конструктор должен иметь специфичный набор параметров.

Деструктор — это функция уничтожения (деинициализации) объекта данного класса. Имя деструктора совпадает с именем класса плюс тильда (~) перед именем. Деструктор не имеет возвращаемого значения и не имеет параметров.

Объекты класса могут создаваться и удаляться двумя способами: с использованием объявления экземпляра класса и с использованием операторов **new** и **delete**.

В случае объявления экземпляра класса объект класса существует в рамках функционального блока (составного оператора, { }). Объект класса размещается в стеке программы.

```
void main() {
     A v1;          // вызов конструктора для v1
     A v2(10);      // вызов конструктора для v2
     A v3(&v1);     // вызов конструктора для v3
     . . .
```

```
     return;
     . . .
     throw;
     . . .
}                    // вызов деструкторов для v1, v1, v3
```

В случае использования операторов **new** и **delete** объект класса создается в динамической памяти и существует до тех пор, пока объект не будет явно удален (оператор delete).

```
void main() {
     A* p1;
     A* p2
     A* p3
     . . .
     p1 = new A;          // вызов конструктора для p1
     p2 = new A(10);      // вызов конструктора для p2
     p3 = new A(p1);      // вызов конструктора для p3
     . . .
     delete p2;           // вызов деструктора для p2
     . . .
     delete p3;           // вызов деструктора для p3
     . . .
     delete p1;           // вызов деструктора для p1
}
```

Указатели на объекты классов

На объекты классов, как и на объекты других типов, можно определять указатели. Затем через указатель можно обращаться к членам класса — переменным и методам. Однако если при обращении через обычную переменную используется символ точка, то для обращения к членам класса через указатель применяется оператор «стрелка» (->):

```
#include <iostream>
#include <string>
using std::string;
using std::cout;
using std::endl;

class Person {
public:
     string name;
     int age;
     void move() {
          cout << name << " is moving" << endl;
     }
};

int main() {
     Person person;
     Person *ptr = &person;
     ptr->name = "Tom";
     ptr->age = 22;
     ptr->move();
```

```
    cout << "Name: " << ptr->name << "\tAge: " << ptr->age <<
endl;
    cout << "Name: " << person.name << "\tAge: " << person.age <<
endl;

    return 0;
}
```

Лабораторное задание

Разработать консольное приложение, в котором должен быть реализован простой класс в соответствии с вариантом задания.

Варианты заданий

1. Класс «Автомобиль» (легковой/грузовой, мощность, объем двигателя, цена и т.д.).
2. Класс «Компьютер» (процессор, объем оперативной памяти, цена и т.д.).
3. Класс «Телефон» (производитель, модель, размер экрана и т.д.).
4. Класс «Книга» (тема, автор, издательство, год выпуска и т.д.).
5. Класс «Журнал» (название, тема, тираж и т.д.).
6. Класс «Студент» (ФИО, ВУЗ, факультет, кафедра, форма обучения и т.д.).
7. Класс «Сотрудник» (ФИО, предприятие, должность, зарплата, стаж и т.д.).
8. Класс «Музыкальное произведение» (название, автор, исполнитель, жанр и т.д.)
9. Класс «Фильм» (название, режиссер, студия, жанр и т.д.)
10. Класс «Населенный пункт» (название, количество жителей, географические координаты, виды общественного транспорта и т.д.).
11. Класс «Предприятие» (название, адрес, направление производства, количество сотрудников и т.д.).
12. Класс «Магазин» (название, адрес, торговая площадь, количество сотрудников и т.д.).
13. Класс «Кинотеатр» (название, адрес, количество залов, репертуар и т.д.).
14. Класс «Телевизор» (производитель, размер экрана, поддерживаемые форматы, цена и т.д.).
15. Класс «Компьютерная игра» (название, жанр, производитель, цена и т.д.).
16. Класс «Одежда» (название, тип, материал, размер, цена и т.д)
17. Класс «Колеса» (типоразмер резины, радиус диска, название резины, название диска и т.д.)
18. Класс «Велосипед» (горный\шоссейный, радиус колес, фирма изготовитель, материал рамы, цена и т.д.)
19. Класс «Отель» (название, количество звезд, этажность, количество номеров и т.д.)

20. Класс «Мебель» (тип, материал, название, фирма изготовитель, габариты и т.д.)

Требования к программе

- Все объекты класса в программе должны создаваться через объявление экземпляра класса на стеке программы.
- В основной функции программы создать объект класса с вызовом конструктора без параметров.
- В основной функции программы создать несколько объектов класса с вызовом конструкторов с параметрами. Должно быть создано столько объектов, сколько реализовано в классе конструкторов с параметрами.
- В основной функции программы объявить массив объектов класса (не более 5).
- В цикле выполнить заполнение массива объектов. Для этого необходимо для каждого объекта в массиве вызвать функцию Input.
- В цикле выполнить печать массива объектов. Для этого необходимо для каждого объекта в массиве вызвать функцию Output.

Требования к классу

- Класс должен содержать конструктор без параметров. В конструкторе должны инициализироваться переменные класса значением по умолчанию.
- Класс должен содержать два или более конструкторов с параметрами. В конструкторах должны инициализироваться переменные класса значениями, которые переданы в параметрах конструктора. Если для какой-либо переменной класса в параметрах конструктора не предусмотрен параметр, то переменная должна инициализироваться значением по умолчанию.
- Класс должен содержать деструктор. В деструкторе все переменные класса должны быть затерты (присвоены недействительные значения).
- Класс должен содержать три или более (в соответствии с вариантом задания) переменных класса. Переменные класса должны быть реализованы в трех секциях: public, protected и private.
- Класс должен содержать одну или более переменную со строковыми данными. Строковые данные должны задаваться только с использованием массива символов (char str1[100] или wchat_t str2[200]).
- В классе должны быть реализованы два метода: Input и Output. Метод Input должен выполнять ввод данных класса с клавиатуры. Метод Output должен печатать на экране данные класса. Методы должны располагаться в секции public.

Требования к оформлению таблиц (ГОСТ 7.32-2001)

1 Таблицы применяют для лучшей наглядности и удобства сравнения показателей. Наименование таблицы, при его наличии, должно отражать ее

содержание, быть точным, кратким. Наименование таблицы следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тире.

(Измененная редакция, Изм. N 1).

2 Таблицу следует располагать в отчете непосредственно после текста, в котором она упоминается впервые, или на следующей странице.

3 На все таблицы должны быть ссылки в отчете. При ссылке следует писать слово "таблица" с указанием ее номера.

4 Таблицу с большим числом строк допускается переносить на другой лист (страницу). При переносе части таблицы на другой лист (страницу) слово "Таблица", ее номер и наименование указывают один раз слева над первой частью таблицы, а над другими частями также слева пишут слова "Продолжение таблицы" и указывают номер таблицы.

Таблицу с большим количеством граф допускается делить на части и помещать одну часть под другой в пределах одной страницы. Если строки и графы таблицы выходят за формат страницы, то в первом случае в каждой части таблицы повторяется головка, во втором случае - боковик. При делении таблицы на части допускается ее головку или боковик заменять соответственно номером граф и строк. При этом нумеруют арабскими цифрами графы и (или) строки первой части таблицы.

Если повторяющийся в разных строках графы таблицы текст состоит из одного слова, то его после первого написания допускается заменять кавычками; если из двух и более слов, то при первом повторении его заменяют словами "То же", а далее - кавычками. Ставить кавычки вместо повторяющихся цифр, марок, знаков, математических и химических символов не допускается. Если цифровые или иные данные в какой-либо строке таблицы не приводят, то в ней ставят прочерк.

5 Цифровой материал, как правило, оформляют в виде таблиц. Пример оформления таблицы приведен на рисунке 1.

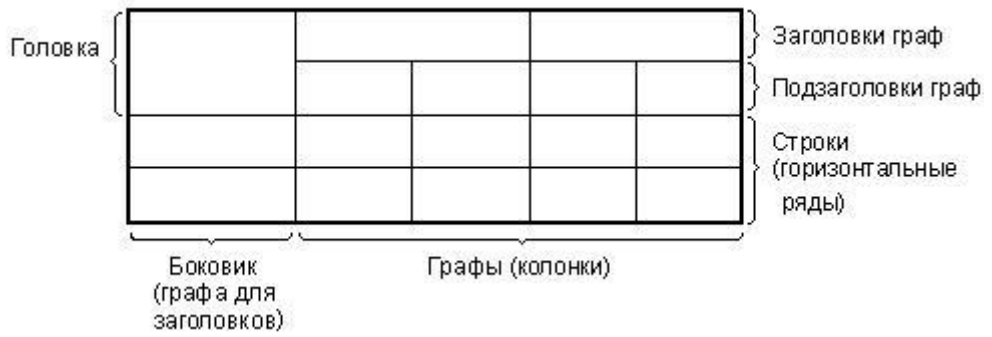

Рисунок 1 — Пример оформления таблицы

6 Таблицы, за исключением таблиц приложений, следует нумеровать арабскими цифрами сквозной нумерацией.

Допускается нумеровать таблицы в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой.

Таблицы каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения.

Если в отчете одна таблица, то она должна быть обозначена "Таблица 1" или "Таблица В.1", если она приведена в приложении В.

4-6 (Измененная редакция, Изм. N 1).

7 Заголовки граф и строк таблицы следует писать с прописной буквы в единственном числе, а подзаголовки граф - со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков таблиц точки не ставят.

8 Таблицы слева, справа и снизу, как правило, ограничивают линиями. Допускается применять размер шрифта в таблице меньший, чем в тексте.

Разделять заголовки и подзаголовки боковика и граф диагональными линиями не допускается.

Горизонтальные и вертикальные линии, разграничивающие строки таблицы, допускается не проводить, если их отсутствие не затрудняет пользование таблицей.

Заголовки граф, как правило, записывают параллельно строкам таблицы. При необходимости допускается перпендикулярное расположение заголовков граф.

Головка таблицы должна быть отделена линией от остальной части таблицы.

9 (Исключен, Изм. N 1).

Требования к оформлению схем

Диаграмма классов (англ. Static Structure diagram) — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их кооперaций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы (классов, типов и т. п.).

Классы можно рассматривать с позиции различных уровней. Как правило, их выделяют три основных: аналитический уровень, уровень проектирования и уровень реализации:

- на уровне анализа класс содержит в себе только набросок общих контуров системы и работает как логическая концепция предметной области или программного продукта.
- на уровне проектирования класс отражает основные проектные решения о распределении информации и планируемой функциональности, объединяя в себе сведения о состоянии и операциях.
- на уровне реализации класс дорабатывается до такого вида, в каком он максимально удобен для воплощения в выбранной среде разработки; при этом не воспрещается опустить в нём те общие свойства, которые не применяются на выбранном языке программирования.

Диаграмма классов является ключевым элементом в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента:

- В верхней части написано имя класса. Имя класса выравнивается по центру и пишется полужирным шрифтом. Имена классов начинаются с заглавной буквы. Если класс абстрактный — то его имя пишется полужирным курсивом.
- Посередине располагаются поля (атрибуты) класса. Они выровнены по левому краю и начинаются с маленькой буквы.
- Нижняя часть содержит методы класса. Они также выровнены по левому краю и пишутся с маленькой буквы.

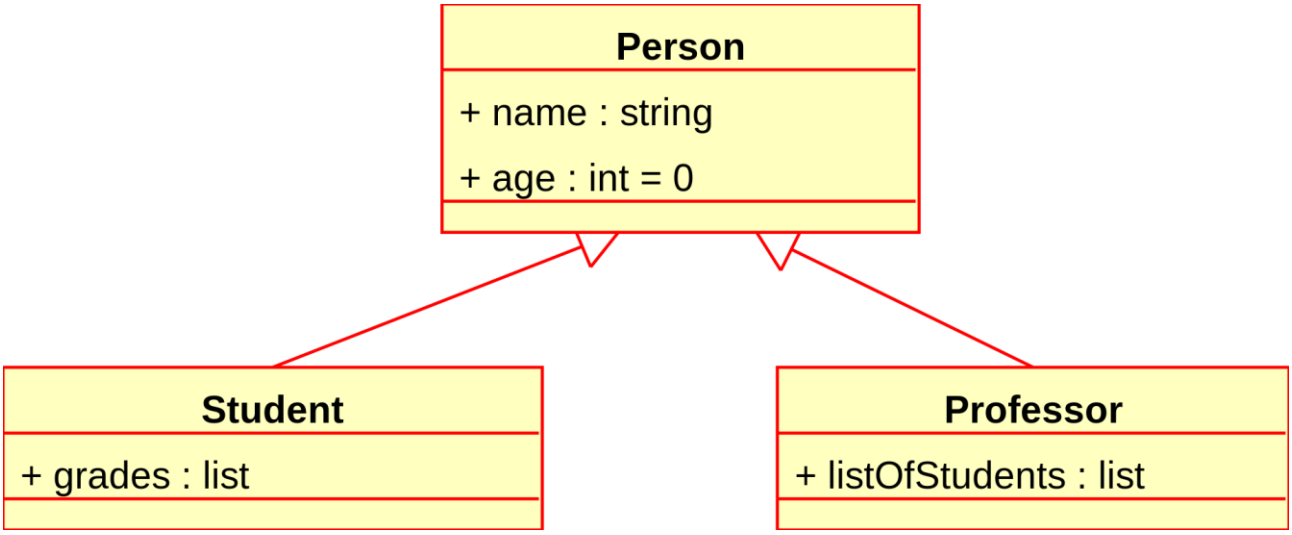

Рисунок 2 — Диаграмма классов, показывающая наследование двух подклассов от одного суперкласса

Групповая работа

Студенты второго курса группы 17ВВ1, проходившие практику в ПГУ, разделились на подгруппы следующим образом:

№ подгруппы	ФИО	Вариант	Оценка
1	Кашлов В. В.	6	

	Сергунин А. А.		
	Харитонов Д. А.		
2	Сенаторова У. В. (досрочное прохождение практики)	6	
3	Тиньгаев В. В. (досрочное прохождение практики)	4	

Студенты второго курса группы 17ВВ1, проходившие практику на предприятии:

№ студента	ФИО	Оценка
1	Сенченко С. И.	
2	Шибаев А. С.	
3	Щеголихин Я. П.	

Подгруппа 1

Студент Сергунин А. А. реализовал следующую часть задания: реализация класса: переменные класса, конструктор, деструктор. реализация методов input, output.

Работа студента заслуживает оценки «_____».

Студент Кашлов В. В. реализовал следующую часть задания: реализация графического интерфейса. тестирование.

Работа студента заслуживает оценки «_____».

Студент Харитонов Д. А. реализовал следующую часть задания: реализация основной функции программы. тестирование.

Работа студента заслуживает оценки «_____».

Подгруппа 2

Студент Сенаторова У. В. проходила практику досрочно. Отчет составлен в соответствии с требованиями.

Руководителем практики была поставлена оценка «_____».

Подгруппа 3

Студент Тиньгаев В. В. проходил практику досрочно. Отчет составлен в соответствии с требованиями.

Руководителем практики была поставлена оценка «_____».

Студент 1

Студент Сенченко С. И. проходил практику «Научно-исполнительская практика» на предприятии. Тема практики «Создание генератора байтовых последовательностей». Отчет составлен в соответствии с требованиями.

Руководителем практики была поставлена оценка «_____».

Студент 2

Студент Шибаев А. С. проходил практику «Научно-исполнительская практика» на предприятии. Тема практики «Создание генератора байтовых последовательностей». Отчет составлен в соответствии с требованиями.

Руководителем практики была поставлена оценка «_____».

Студент 3

Студент Щеголихин Я. П. проходил практику «Научно-исполнительская практика» на предприятии. Тема практики «Исследование возможностей распознания библиотекой tesseract». Отчет составлен в соответствии с требованиями.

Руководителем практики была поставлена оценка «_____».

Студенты проявили свои знания в предметных областях инженерных направлений, а также знания и навыки, полученные за два года обучения в бакалавриате.

Все студенты выполнили возложенные на них задачи при разделении их внутри подгруппы (бригады).

Заключение

В результате прохождения учебной практики выполнена разработка учебно-методических указаний для прохождения учебной практики бакалавров 2 курса обучения. В их состав входят 20 вариантов заданий на анализ предметной области и написании программы в ООП стиле на языке C++, а также разработка псевдографического интерфейса. Ведение научно-исполнительской практики у студентов бакалавриата 2-ого курса происходило дистанционно.

При составлении учебно-методических указаний учитывалось содержание учебных планов бакалавров 1, 2 курсов. При определении заданий учитывалась необходимость постановки групповой работы, чтобы в процессе ее решения бакалавры выполнили исследование предложенной предметной области, выделили ее основные аспекты, составили диаграмму классов, написали программу на ОПП языке C++ и проверили корректность ее работы. В процесс выполнения этой объемной работы бакалавры второго курса расширили познания в сфере анализа и декомпозиции предметных областей, познакомились с основными парадигмами ООП, а также получили опыт в проведении тестирования.

Использованные в ходе выполнения учебной практики подходы к групповому разделению задач и делегированию ролей потребовали высокой координации деятельности, особенно на этапе консолидации результатов и их анализа.

Цели выполнения учебной практики считаю достигнутыми.

Список использованных источников

1. Грэди Буч — Объектно-ориентированный анализ и проектирование с примерами приложений на C++
2. Герберт Шилдт — Самоучитель C++
3. Стенли Липпман — Основы программирования на C++
4. Бьерн Страуструп — Язык программирования C++
5. Столяров А. В. — Введение в язык Си++
6. http://docs.cntd.ru/document/gost-7-32-2001-sibid
7. https://en.wikipedia.org/wiki/Class_diagram
8. https://prog-cpp.ru/uml-classes/

Приложение А. Пример оформления титульных листов

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО "ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника»

"___" _____ 20 г.

Заведующий кафедрой

_____М.А. Митрохин

ОТЧЕТ ПО УЧЕБНОЙ (НАУЧНО-ИСПОЛНИТЕЛЬСКОЙ) ПРАКТИКЕ

(2018/2019 учебный год)

Нешко Дарья Олеговна

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения 2 семестр 4

Период прохождения практики с 8.07.2019 по 21.07.2019

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., Митрохин М. А.

(должность, ученая степень, ученое звание, Ф.И.О.)

Научный руководитель к.т.н., Карамышева Н. С.

(должность, ученая степень, ученое звание)

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО "ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника»

"___" _____ 20 г.

Заведующий кафедрой

_____М.А. Митрохин

ИНДИВИДУАЛЬНЫЙ ПЛАН ПРОХОЖДЕНИЯ УЧЕБНОЙ (НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ) ПРАКТИКИ

(2018/2019 учебный год)

Нешко Дарья Олеговна

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения 2 семестр 4

Период прохождения практики с 08.07.2019 по 21.07.2019

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., Митрохин М. А.

(должность, ученая степень, ученое звание, Ф.И.О.)

Научный руководитель к.т.н., Карамышева Н. С.

(должность, ученая степень, ученое звание)

№ п/п	Планируемая форма работы во время практики	Количество часов	Календарные сроки проведения работы	Подпись руководителя практики от вуза
1	Выбор темы и разработка индивидуального плана проведения работ	6	08.07.2019 г.	
2	Подбор и изучение материала по теме работы	24	09.07.2019 г.-11.07.2019 г.	
3	Разработка программной части	24	12.07.2019 г.-13.07.2019 г.	
4	Разработка графического интерфейса	12	14.07.2019 г.	
5	Отладка программы	8	15.07.2019 г.	
6	Получение и анализ результатов	7	16.07.2019 г.	
7	Консультирование с руководителем о проделанной работе	1	17.07.2019 г.	
8	Оформление отчёта	25	18.07.2019 г.-20.07.2019 г.	
9	Консультирование с руководителем о подготовленном отчете	1	21.07.2019 г.	
	Общий объём часов	108	08.07.2019 г.-21.07.2019 г.	

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО "ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЗЫВ
О ПРОХОЖДЕНИИ УЧЕБНОЙ (НАУЧНО-ИСПОЛНИТЕЛЬСКОЙ) ПРАКТИКИ

(2018/2019 учебный год)

Нешко Дарья Олеговна

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения 2 семестр 4

Период прохождения практики с 08.07.2019 по 21.07.2019

Кафедра «Вычислительная техника»

В процессе выполнения научно-исследовательской работы Нешко Д.О. решала следующие задачи: поиск и анализ методологий программирования, анализ объектно-ориентированного программирования, сравнение существующих методов разработки классов в языке программирования C++ .

За период выполнения научно-исполнительской практики были освоены основные понятия и технологии объектно-ориентированного программирования на примере простых классов, изучены основные инструменты среды разработки Microsoft Visual Studio 2017. И получены следующие результаты: было разработано консольное приложение, содержащее класс в соответствии с вариантом. Во время выполнения работы Нешко Д.О. показала себя ответственным, добросовестным учеником, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике, программированию и алгоритмизации .

За выполнение работы Нешко Д.О. заслуживает оценки «отлично».

Научный руководитель к.т.н., Карамышева Н. С. « 21 » июля 2019 г.