

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Уральский государственный экономический университет»

**Одобрена**  
на заседании кафедры

10.01.2020 г.

протокол № 6

Зав. кафедрой Сурнина Н.М.

**Утверждена**

Советом по учебно-методическим вопросам  
и качеству образования

15 января 2020 г.

протокол № 5

Председатель



(подпись)

Карх Д.А.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

Наименование дисциплины	Объектно-ориентированное программирование
Направление подготовки	02.03.03 Математическое обеспечение и администрирование информационных систем
Профиль	Разработка и администрирование информационных систем
Форма обучения	очная
Год набора	2020

Разработана:  
Доцент, к.э.н.  
Кислицын Е.В.

Екатеринбург  
2020 г.

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>3</b>
<b>1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ</b>	<b>3</b>
<b>2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП</b>	<b>3</b>
<b>3. ОБЪЕМ ДИСЦИПЛИНЫ</b>	<b>3</b>
<b>4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП</b>	<b>3</b>
<b>5. ТЕМАТИЧЕСКИЙ ПЛАН</b>	<b>4</b>
<b>6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ</b>	<b>5</b>
<b>7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ</b>	<b>7</b>
<b>8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ</b>	<b>11</b>
<b>9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ</b>	<b>11</b>
<b>10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ</b>	<b>11</b>
<b>11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ</b>	<b>12</b>

## ВВЕДЕНИЕ

Рабочая программа дисциплины является частью основной профессиональной образовательной программы высшего образования - программы бакалавриата, разработанной в соответствии с ФГОС ВО

ФГОС ВО	Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата) (приказ Минобрнауки России от 23.08.2017 г. № 809)
ПС	

### 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Получение знаний, умений и навыков в области разработки программного обеспечения на языках высокого уровня.

### 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к вариативной части учебного плана.

### 3. ОБЪЕМ ДИСЦИПЛИНЫ

Промежуточный контроль	Часов				З.е.
	Всего за семестр	Контактная работа (по уч.зан.)		Самостоятельная работа в том числе подготовка контрольных и курсовых	
		Всего	Лабораторные		
Семестр 3					
Зачет	108	56	56	52	3
Семестр 4					
Экзамен, Курсовая работа	180	54	54	99	5
	288	110	110	151	8

### 4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП

В результате освоения ОПОП у выпускника должны быть сформированы компетенции, установленные в соответствии ФГОС ВО.

Общепрофессиональные компетенции (ОПК)

Шифр и наименование компетенции	Индикаторы достижения компетенций
ОПК-3 Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	ИД-1.ОПК-3 Знать: основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов. Уметь: решать стандартные задачи профессиональной деятельности с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности. Иметь навыки: иметь практические навыки разработки современного программного обеспечения с учетом требований информационной безопасности.

## Профессиональные компетенции (ПК)

Шифр и наименование компетенции	Индикаторы достижения компетенций
<b>производственно-технологический</b>	
ПК-1 Способен применять современные информационные технологии при проектировании и реализации программного обеспечения для решения задач в различных предметных областях	ИД-1.ПК-1 Знать: современные технологии проектирования и производства программного продукта. Уметь: использовать подобные технологии при создании программных продуктов, анализировать исходные данные и применять методы моделирования информационных процессов. Иметь навыки: разработки моделей и информационных процессов; проектирования структур и баз данных.
ПК-5 Способен использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений	ИД-1.ПК-5 Знать: основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений. Уметь: разрабатывать и проектировать программное обеспечение; использовать паттерны проектирования и программирования. Иметь навыки: объектно-ориентированного анализа и проектирования программных систем и систем автоматического управления.

## 5. ТЕМАТИЧЕСКИЙ ПЛАН

Тема	Часов						
	Наименование темы	Всего часов	Контактная работа (по уч.зан.)			Самост. работа	Контроль самостоятельной работы
			Лекции	Лабораторные	Практические занятия		
<b>Семестр 3</b>		<b>108</b>					
Тема 1.	Основы языка программирования Java.	18		12		6	
Тема 2.	Классы и объекты. Статический и нестатический контекст класса. Конструкторы, блоки инициализации.	14		8		6	
Тема 3.	Инкапсуляция: самодостаточные объекты, объект как черный ящик. Модификаторы доступа.	14		8		6	
Тема 4.	Наследование: расширение поведения.	14		8		6	
Тема 5.	Абстракция: от общего к частному через наследование. Абстрактные классы и интерфейсы. Модификаторы abstract и final.	11		4		7	
Тема 6.	Полиморфизм: множество реализаций.	11		4		7	
Тема 7.	Агрегация и композиция: сложные объекты. Объекты из объектов. Расширение поведения без наследования.	11		4		7	

Тема 8.	Статическая типизация языка Java. Приведение типов. Обобщенные типы как решение проблемы типобезопасности.	15	8	7		
Семестр 4		153				
Тема 9.	Коллекции объектов, сравнение объектов и компараторы.	19	8	11		
Тема 10.	Неизменяемые объекты.	15	4	11		
Тема 11.	Работа со строками и потоками ввода/вывода.	15	4	11		
Тема 12.	Обработка ошибок и исключения.	15	4	11		
Тема 13.	Работа с сетью через Socket.	15	4	11		
Тема 14.	Многопоточное программирование.	19	8	11		
Тема 15.	Графический интерфейс.	24	12	12		
Тема 16.	Примеры разработки объектно-ориентированных программных систем на языке Java.	31	10	21		

## 6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ

Раздел/Тема	Вид оценочного средства	Описание оценочного средства	Критерии оценивания
Текущий контроль (Приложение 4)			
Тема 1-4.	Тест (приложение 4)	Тест содержит 29 вопросов	10 баллов
Тема 5-8.	Практическая работа (приложение 4)	Практическая работа содержит 4 задания на программирование.	10 баллов
Тема 9-16.	Практическая работа (приложение 4)	Практическая работа содержит 1 задание на программирование.	10 баллов
Промежуточный контроль (Приложение 5)			
4 семестр (Эк)	Экзаменационный билет (приложение 5)	Экзаменационный билет содержит два теоретических вопроса и одно практическое задание.	Теоретические вопросы - по 25 баллов, практическое задание - 50 баллов.
4 семестр (КР)	Курсовая работа (приложение 7)	Курсовая работа содержит две главы, содержит описание разработки программного комплекса.	100 баллов

3 семестр (За)	Зачет (Приложение 5)	Билет содержит два теоретических вопроса и одно практическое задание.	Теоретические вопросы - по 25 баллов, практическое задание - 50 баллов.
----------------	----------------------	---	---

### **ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ**

Показатель оценки освоения ОПОП формируется на основе объединения текущей и промежуточной аттестации обучающегося.

Показатель рейтинга по каждой дисциплине выражается в процентах, который показывает уровень подготовки студента.

Текущая аттестация. Используется 100-балльная система оценивания. Оценка работы студента в течении семестра осуществляется преподавателем в соответствии с разработанной им системой оценки учебных достижений в процессе обучения по данной дисциплине.

В рабочих программах дисциплин и практик закреплены виды текущей аттестации, планируемые результаты контрольных мероприятий и критерии оценки учебных достижений.

В течение семестра преподавателем проводится не менее 3-х контрольных мероприятий, по оценке деятельности студента. Если посещения занятий по дисциплине включены в рейтинг, то данный показатель составляет не более 20% от максимального количества баллов по дисциплине.

Промежуточная аттестация. Используется 5-балльная система оценивания. Оценка работы студента по окончанию дисциплины (части дисциплины) осуществляется преподавателем в соответствии с разработанной им системой оценки достижений студента в процессе обучения по данной дисциплине. Промежуточная аттестация также проводится по окончанию формирования компетенций.

Порядок перевода рейтинга, предусмотренных системой оценивания, по дисциплине, в пятибалльную систему.

Высокий уровень – 100% - 70% - отлично, хорошо.

Средний уровень – 69% - 50% - удовлетворительно.

Показатель оценки	По 5-балльной системе	Характеристика показателя
100% - 85%	отлично	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на высоком уровне
84% - 70%	хорошо	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов.  Могут быть допущены недочеты, исправленные студентом самостоятельно в процессе работы (ответа и т.д.)
69% - 50%	удовлетворительно	обладают общими теоретическими знаниями, умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на среднем уровне. Допускаются ошибки, которые студент затрудняется исправить самостоятельно.
49 % и менее	неудовлетворительно	обладают не полным объемом общих теоретическими знаниями, не умеют самостоятельно применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Не сформированы умения и навыки для решения
100% - 50%	зачтено	характеристика показателя соответствует «отлично», «хорошо», «удовлетворительно»
49 % и менее	не зачтено	характеристика показателя соответствует «неудовлетворительно»

## 7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 7.2 Содержание практических занятий и лабораторных работ

Тема 1. Основы языка программирования Java.

Язык программирования Java. Первая программа на Java. Первая программа в IntelliJ IDEA. Структура программы. Переменные и константы. Типы данных. Консольный ввод/вывод в Java. Арифметические операции. Побитовые операции. Условные выражения. Операции присваивания и приоритет операций. Преобразования базовых типов данных. Условные конструкции. Циклы. Массивы. Методы. Параметры методов. Оператор return. Результат метода.

Тема 2. Классы и объекты. Статический и нестатический контекст класса. Конструкторы, блоки инициализации.

Основы классов. Общая форма класса. Простой класс. Объявление классов. Оператор new. Присваивание переменных объектных ссылок. Методы. Добавление метода к классу. Возвращение значения. Добавление метода, принимающего параметры. Конструкторы. Конструкторы с параметрами. Ключевое слово this. Сборка мусора.

Тема 3. Инкапсуляция: самодостаточные объекты, объект как черный ящик. Модификаторы доступа.

Перегрузка методов. Перегрузка конструкторов. Использование объектов в качестве параметров. Возврат объектов. Рекурсия. Управление доступом. Ключевые слова static, final. Массивы. Класс String. Список аргументов переменной длины. Модификаторы доступа public, private. Геттеры и

Тема 4. Наследование: расширение поведения.

Основы наследования. Доступ к членам и наследование. Модификатор protected. Ключевое слово super. Создание многоуровневой иерархии. Порядок вызова конструкторов. Переопределение методов. Динамическая диспетчеризация методов.

Тема 5. Абстракция: от общего к частному через наследование. Абстрактные классы и интерфейсы. Модификаторы abstract и final.

Абстрактные классы. Ключевое слово final. Класс Object. Определение интерфейса. Реализация интерфейсов. Доступ к реализациям через ссылки на интерфейсы. Вложенные интерфейсы. Возможность расширения интерфейсов.

Тема 6. Полиморфизм: множество реализаций.

Восходящее преобразование. Связывание "метод-вызов". Получение нужного результата. Расширяемость. Проблема: переопределение закрытых методов. Проблема: поля и статические методы. Конструкторы и полиморфизм. Ковариантность возвращаемых типов. Наследование при проектировании.

Тема 7. Агрегация и композиция: сложные объекты. Объекты из объектов. Расширение поведения без наследования.

Создание внутренних классов. this и new. Внутренние классы и восходящее преобразование. Внутренние классы в методах и областях действия. Анонимные внутренние классы. "Фабричный метод". Вложенные классы. Наследование от внутренних классов.

Тема 8. Статическая типизация языка Java. Приведение типов. Обобщенные типы как решение проблемы типобезопасности.

Простые обобщения (кортежи, стек, RandomList). Обобщенный класс с двумя параметрами. Общая форма обобщенного класса. Ограниченные типы. Использование шаблонов аргументов. Обобщенные интерфейсы. Обобщенные методы. Базовые типы и унаследованный код. Иерархии обобщенных классов. Выведение типов и обобщения. Очистка. Ограничения обобщений.

Тема 9. Коллекции объектов, сравнение объектов и компараторы.

Основные концепции. Добавление групп элементов. List. Итераторы. LinkedList. Стек. Множество. Map. Очередь. PriorityQueue. Collection и Iterator. Foreach и итераторы. Идиома "Метод-Адаптер". Полная таксономия контейнеров. Функциональность Collection.



<p>Тема 10. Неизменяемые объекты.</p> <p>Постоянство строк. Перегрузка+ и StringBuilder. Непреднамеренная рекурсия. Форматирование вывода. Регулярные выражения. StringTokenizer.</p>
<p>Тема 11. Работа со строками и потоками ввода/вывода.</p> <p>Класс File. Ввод и вывод. Добавление атрибутов и интерфейсов. Классы Reader и Writer. Типичное использование потоков ввода-вывода. Средства чтения и записи файлов. Стандартный ввод-вывод. Управление процессами. Новый ввод-вывод. Сжатие данных. Сериализация объектов.</p>
<p>Тема 12. Обработка ошибок и исключения.</p> <p>Основные концепции. Основные исключения. Перехват исключений. Создание собственных исключений. Вывод информации об исключениях. Спецификация исключений. Перехват любого типа исключений. Стандартные исключения Java. Блок finally. Ограничения исключений. Конструкторы. Отожествление исключений.</p>
<p>Тема 13. Работа с сетью через Socket.</p> <p>Основы работы с сетью. Сетевые классы и интерфейсы. Класс InetAddress. Клиентские сокеты TCP/IP. Файлы cookie. Серверные сокеты TCP/IP. Дейтаграммы.</p>
<p>Тема 14. Многопоточное программирование.</p> <p>Основы построения многопоточных программ. Класс Thread. Возвращение значений из задач. Группы потоков. Совместное использование ресурсов. Завершение задач. Завершение при блокировке. Взаимодействие между задачами. Взаимная блокировка. Моделирование.</p>
<p>Тема 15. Графический интерфейс.</p> <p>Апплеты. Основы Swing. Создание кнопки. Перехват событий. Управление расположением компонентов. Модель событий библиотеки Swing. Компоненты Swing. Введение в JavaFX. FXML. Панели компоновки. Элементы управления. Введение в разработку Android-приложений.</p>
<p>Тема 16. Примеры разработки объектно-ориентированных программных систем на языке Java.</p> <p>Взаимодействие с базой данных. Разработка многопользовательского чата. Разработка игры "Морской бой".</p>

### 7.3. Содержание самостоятельной работы

<p>Тема 1. Основы языка программирования Java.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 2. Классы и объекты. Статический и нестатический контекст класса. Конструкторы, блоки инициализации.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 3. Инкапсуляция: самодостаточные объекты, объект как черный ящик. Модификаторы доступа.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ.</p>
<p>Тема 4. Наследование: расширение поведения.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 5. Абстракция: от общего к частному через наследование. Абстрактные классы и интерфейсы. Модификаторы abstract и final.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 6. Полиморфизм: множество реализаций.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>

<p>Тема 7. Агрегация и композиция: сложные объекты. Объекты из объектов. Расширение поведения без наследования.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 8. Статическая типизация языка Java. Приведение типов. Обобщенные типы как решение проблемы типобезопасности.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 9. Коллекции объектов, сравнение объектов и компараторы.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 10. Неизменяемые объекты.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 11. Работа со строками и потоками ввода/вывода.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 12. Обработка ошибок и исключения.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 13. Работа с сетью через Socket.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю. Написание курсовой работы.</p>
<p>Тема 14. Многопоточное программирование.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю. Написание курсовой работы.</p>
<p>Тема 15. Графический интерфейс.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю. Написание курсовой работы.</p>
<p>Тема 16. Примеры разработки объектно-ориентированных программных систем на языке Java.</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю. Написание курсовой работы.</p>

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену  
Приложение 1.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену  
Приложение 2.

7.3.3. Перечень курсовых работ  
Приложение 3.

7.4. Электронное портфолио обучающегося  
Размещается Курсовая работа

7.5. Методические рекомендации по выполнению контрольной работы  
Не предусмотрено.

7.6 Методические рекомендации по выполнению курсовой работы  
Приложение 7.

## **8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ**

### ***По заявлению студента***

В целях доступности освоения программы для лиц с ограниченными возможностями здоровья при необходимости кафедра обеспечивает следующие условия:

- особый порядок освоения дисциплины, с учетом состояния их здоровья;
- электронные образовательные ресурсы по дисциплине в формах, адаптированных к ограничениям их здоровья;
- изучение дисциплины по индивидуальному учебному плану (вне зависимости от формы обучения);
- электронное обучение и дистанционные образовательные технологии, которые предусматривают возможности приема-передачи информации в доступных для них формах.
- доступ (удаленный доступ), к современным профессиональным базам данных и информационным справочным системам, состав которых определен РПД.

## **9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

### **Сайт библиотеки УрГЭУ**

<http://lib.usue.ru/>

### **Основная литература:**

1. Кислицын Е. В., Шишков Е. И.. Разработка приложений на языке Java [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2017. - 86 с. – Режим доступа: <http://lib.usue.ru/resource/limit/ump/17/p488938.pdf>

2. Плещев В. В., Шишков Е. И.. Основы программирования на языках C++ и C# с примерами и упражнениями [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2018. - 286 с. – Режим доступа: <http://lib.usue.ru/resource/limit/ump/18/p490708.pdf>

### **Дополнительная литература:**

1. Шилдт Г., Гузикевич А. Г.. Java 8: руководство для начинающих. - Москва: Вильямс, 2017. - 712 с.

2. Хорев П. Б.. Объектно-ориентированное программирование с примерами на C# [Электронный ресурс]: учебное пособие для студентов вузов, обучающихся по направлениям 01.03.02 "Прикладная математика и информатика" и 09.00.00 "Информатика и выч. техника". - Москва: ФОРУМ: ИНФРА-М, 2016. - 200 с. – Режим доступа: <http://znanium.com/go.php?id=529350>

3. Жеребцов А. С., Молодецкая С. Ф.. Объектно-ориентированный анализ и программирование [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2014. - 126 с. – Режим доступа: <http://lib.usue.ru/resource/limit/ump/14/p482025.pdf>

## **10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

### **Перечень лицензионное программное обеспечение:**

Astra Linux Common Edition. Договор № 1 от 13 июня 2018, акт от 17 декабря 2018. Срок действия лицензии - без ограничения срока.

МойОфис стандартный. Соглашение № СК-281 от 7 июня 2017. Дата заключения - 07.06.2017. Срок действия лицензии - без ограничения срока.

Libre Office. Лицензия GNU LGPL. Срок действия лицензии - без ограничения срока.

Microsoft Visual Studio Community. Лицензия для образовательных учреждений. Срок действия лицензии - без ограничения срока.

MySQL Community Server. Стандартная общественная лицензия GNU (GPL). Срок действия лицензии - без ограничения срока.

Microsoft Visio. Акт предоставления прав № Tr020776 от 07.04.2017. Срок действия лицензии - 07.02.2020.

Notepad++. Лицензия GNU General Public License. Срок действия лицензии - без ограничения срока.

**Перечень информационных справочных систем, ресурсов информационно-телекоммуникационной сети «Интернет»:**

Справочно-правовая система Гарант. Договор № 58419 от 22 декабря 2015. Срок действия лицензии - без ограничения срока

**11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

Реализация учебной дисциплины осуществляется с использованием материально-технической базы УрГЭУ, обеспечивающей проведение всех видов учебных занятий и научно-исследовательской и самостоятельной работы обучающихся:

Специальные помещения представляют собой учебные аудитории для проведения всех видов занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду УрГЭУ.

Все помещения укомплектованы специализированной мебелью и оснащены мультимедийным оборудованием спецоборудованием (информационно-телекоммуникационным, иным компьютерным), доступом к информационно-поисковым, справочно-правовым системам, электронным библиотечным системам, базам данных действующего законодательства, иным информационным ресурсам служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа презентации и другие учебно-наглядные пособия обеспечивающие тематические иллюстрации

### 7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену

#### К зачету

1. Базовые типы данных. Примитивные типы, ссылочные типы.
2. Логические операторы. Приоритет выполнения. Сокращенные версии логических операторов `&&`, `||` и их отличия от `|`, `&`.
3. Арифметические операторы. Операторы инкремента, декремента, приоритет операций.
4. Приведение типов (явное и автоматическое). Константы и переменные.
5. Операторы ветвления. Условный оператор. Тернарный условный оператор.
6. Операторы ветвления. Оператор выбора. Его отличия от условного оператора.
7. Циклы. Операторы цикла и их отличия.
8. Массивы. Инициализация и индексация массива, многомерные массивы.
9. Понятие класса и объекта. Определение понятий. Статический и не статический контекст класса.
10. Понятие класса и объекта. Члены класса. Методы, поля, конструкторы, блоки инициализации. Ключевые слова `abstract` и `final`.
11. Понятие класса и объекта. Внутренние классы, вложенные классы, анонимные классы, определение, особенности.
12. Основополагающие принципы ООП. Инкапсуляция. Средства реализации инкапсуляции. Модификаторы доступа.
13. Основополагающие принципы ООП. Наследование. Управление наследованием.
14. Основополагающие принципы ООП. Полиморфизм. Средства реализации полиморфизма.
15. Сигнатура метода. Понятия перегрузки и переопределения.

#### К экзамену

1. Базовые типы данных. Примитивные типы, ссылочные типы.
2. Логические операторы. Приоритет выполнения. Сокращенные версии логических операторов `&&`, `||` и их отличия от `|`, `&`.
3. Арифметические операторы. Операторы инкремента, декремента, приоритет операций.
4. Приведение типов (явное и автоматическое). Константы и переменные.
5. Операторы ветвления. Условный оператор. Тернарный условный оператор.
6. Операторы ветвления. Оператор выбора. Его отличия от условного оператора.
7. Циклы. Операторы цикла и их отличия.
8. Массивы. Инициализация и индексация массива, многомерные массивы.
9. Понятие класса и объекта. Определение понятий. Статический и не статический контекст класса.
10. Понятие класса и объекта. Члены класса. Методы, поля, конструкторы, блоки инициализации. Ключевые слова `abstract` и `final`.
11. Понятие класса и объекта. Внутренние классы, вложенные классы, анонимные классы, определение, особенности.
12. Основополагающие принципы ООП. Инкапсуляция. Средства реализации инкапсуляции. Модификаторы доступа.
13. Основополагающие принципы ООП. Наследование. Управление наследованием.
14. Основополагающие принципы ООП. Полиморфизм. Средства реализации полиморфизма.
15. Сигнатура метода. Понятия перегрузки и переопределения.

16. Понятие класса и интерфейса: абстрактные классы, абстрактные методы. Отличие абстрактного класса от интерфейса;
17. Интерфейсы: определение, реализация, наследование;
18. Перечисления (Enum): определение, реализация, особенности;
19. Дженерики: Определение, реализация, примеры;
20. Дженерики: параметризованные методы, маски (wildcards). Примеры;
21. Исключения. Определение исключения, иерархия исключений;
22. Исключения. Проверяемые исключения, определение, особенности, пример реализации собственных исключений;
23. Исключения. Непроверяемые исключения, определение, особенности, пример реализации собственных исключений;
24. Исключения. Error. Определение, особенности, примеры;
25. Исключения. Обработка исключений. Исключения и наследование.
26. Потоки ввода-вывода. Основные типы потоков ввода-вывода, основные методы, различия;
27. Символьные потоки ввода-вывода. Основные представители, описание;
28. Байтовые потоки ввода/вывода. Основные представители, описание;
29. Сетевое взаимодействие с помощью Socket. Описание, механизм работы, примеры использования.
30. Сериализация и десериализация объектов. Определение, условия, реализация, внедрение в процесс сериализации;
31. Методы класса Object hashCode() и equals(). Описание, правила реализации;
32. Интерфейсы Comparator и Comparable. Описание, различия, примеры использования;
33. Collection framework. Коллекции типа Set. Описание, представители. Механизм работы, различия реализаций;
34. Collection framework. Коллекции типа List. Описание, представители. Механизм работы, различия реализаций;
35. Collection framework. Коллекции типа Map. Описание, представители. Механизм работы, различия реализаций;
36. Collection framework. Коллекции типа Queue/Deque. Описание, представители. Механизм работы, различия;
37. Потоки исполнения. Понятия потока и процесса. Жизненный цикл потока;
38. Потоки исполнения. Способы создания потоков исполнения. Класс Thread и его методы start(), run(), sleep(), yield());
39. Потоки исполнения. Прерывание потоков;
40. Потоки исполнения. Взаимодействие потоков. Понятия Мьютекс, Монитор, Критическая секция и их реализация в языке Java.
41. Потоки исполнения. Методы wait(), notify(), notifyAll() класса Object.
42. Потоки исполнения. Синхронизация потоков;
43. Потоки исполнения. Атомарность операций. Атомарные типы данных;
44. Потоки исполнения. Состояние гонки (Data Race). Определение, причины, пути решения.
45. Потоки исполнения. Взаимная блокировка. Определение, причины, пути решения.

### 7.3.2. Практические задания по дисциплине для самостоятельной подготовки к экзаменам

#### К зачету

1. Заполните массив случайными числами и выведите максимальное, минимальное и среднее значение. Для генерации случайного числа используйте метод `Math.random()`, который возвращает значение в промежутке  $[0, 1]$ .
2. Реализуйте алгоритм сортировки пузырьком для сортировки массива.
3. Напишите программу, которая выводит на консоль простые числа в промежутке от  $[2, 100]$ . Используйте для решения этой задачи оператор `"%"` (остаток от деления) и циклы.
4. Дан массив целых чисел и ещё одно целое число. Удалите все вхождения этого числа из массива (пропусков быть не должно).
5. Создайте класс, который описывает вектор (в трёхмерном пространстве). У него должны быть:

- конструктор с параметрами в виде списка координат  $x, y, z$
- метод, вычисляющий длину вектора. Корень можно посчитать с помощью `Math.sqrt()`:

$$\text{Math.sqrt}(x^2+y^2+z^2)$$

- метод, вычисляющий скалярное произведение:

$$x_1x_2+y_1y_2+z_1z_2$$

- метод, вычисляющий векторное произведение с другим вектором:

$$(y_1z_2-z_1y_2, z_1x_2-x_1z_2, x_1y_2-y_1x_2)$$

- метод, вычисляющий угол между векторами (или косинус угла): косинус угла между векторами равен скалярному произведению векторов, деленному на произведение модулей (длин) векторов:

$$(a, b)/|a| \cdot |b|$$

- методы для суммы и разности:

$$(x_1+x_2, y_1+y_2, z_1+z_2)$$

$$(x_1-x_2, y_1-y_2, z_1-z_2)$$

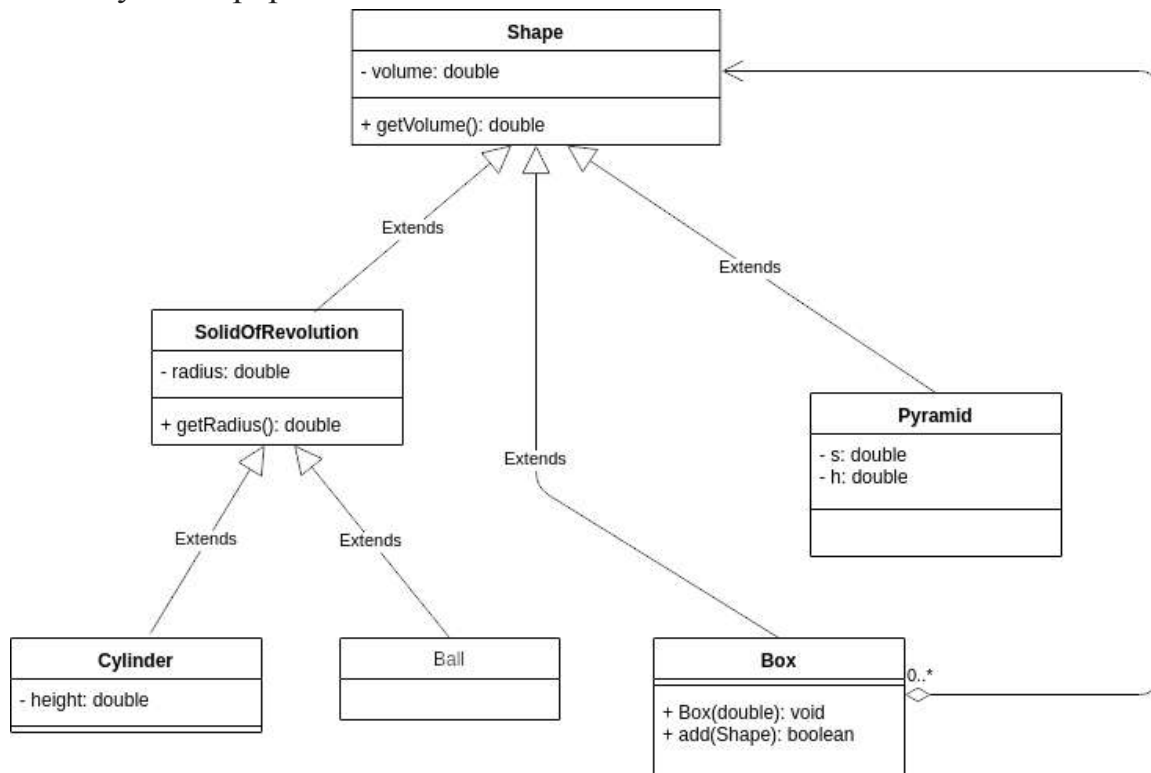
- статический метод, который принимает целое число N, и возвращает массив случайных векторов размером N. Если метод возвращает вектор, то он должен возвращать новый объект, а не менять базовый. То есть, нужно реализовать шаблон "Неизменяемый объект"

6. Напишите класс, конструктор которого принимает два массива: массив значений и массив весов значений. Класс должен содержать метод, который будет возвращать элемент из первого массива случайным образом, с учётом его веса. Пример: Дан массив [1, 2, 3], и массив весов [1, 2, 10]. В среднем, значение «1» должно возвращаться в 2 раза реже, чем значение «2» и в десять раз реже, чем значение «3».
7. Напишите метод, который проверяет, входит ли в массив заданный элемент или нет. Используйте перебор и двоичный поиск для решения этой задачи. Сравните время выполнения обоих решений для больших массивов (например, 100000000 элементов).
8. Найдите корень уравнения

$$\cos(x^5) + x^4 - 345.3 * x - 23 = 0$$

на отрезке [0; 10] с точностью по x не хуже, чем 0.001. Известно, что на этом промежутке корень единственный. Используйте для этого метод деления отрезка пополам (и рекурсию).

9. Реализуйте иерархию классов:



Класс **Box** является контейнером, он может содержать в себе другие фигуры. Метод *add()* принимает на вход Shape. Нужно добавлять новые фигуры до тех пор, пока для них хватает места в Box (будем считать только объём, игнорируя форму. Допустим, мы переливаем жидкость). Если места для добавления новой фигуры не хватает, то метод должен вернуть **false**.

10. Реализуйте ту же иерархию классов, но сделав некоторые классы абстрактными.



11. Реализуйте ту же иерархию классов, но используя интерфейсы.
12. Постройте частотный словарь букв русского (или английского) алфавита. Опустим проблему выбора и анализа корпуса языка, достаточно будет взять текст небольшой длины).

### К экзамену

1. Напишите класс `BaseConverter` для конвертации из градусов по Цельсию в Кельвины, Фаренгейты, и так далее. У метода должен быть метод `convert`, который и делает конвертацию.
2. Напишите свой класс `StringBuilder` с поддержкой операции `undo`. Для этого делегируйте все методы стандартному `StringBuilder`, а в собственном классе храните список всех операций для выполнения `undo()`. Это будет реализацией шаблона «Команда».
3. Напишите свой класс `StringBuilder`, с возможностью оповещения других объектов об изменении своего состояния. Для этого делегируйте все методы стандартному `StringBuilder`, а в собственном классе реализуйте шаблон проектирования «Наблюдатель».
4. Напишите метод `filter`, который принимает на вход массив (любого типа) и реализацию интерфейса `Filter` с методом `apply(Object o)`, чтобы убрать из массива лишнее. Проверьте как он работает на строках или других объектах.
5. Напишите метод `fill`, который принимает массив объектов, и реализацию интерфейса `Function` (или своего). Метод `fill` должен заполнить массив, получая новое значение по индексу с помощью реализации интерфейса `Function`.
6. Постройте частотный словарь слов русского (или английского) алфавита. Опустим проблему выбора и анализа корпуса языка, достаточно будет взять текст небольшой длины).
7. Напишите метод, который на вход получает коллекцию объектов, а возвращает коллекцию уже без дубликатов.
8. Напишите метод, который добавляет 1000000 элементов в `ArrayList` и `LinkedList`. Напишите еще один метод, который выбирает из заполненного списка элемент наугад 100000 раз. Замерьте время, которое потрачено на это. Сравните результаты и предположите, почему они именно такие.
9. Написать итератор по массиву.
10. Написать итератор по двумерному массиву.
11. Дан итератор. Метод `next()` возвращает либо `String`, либо итератор такой же структуры (то есть который опять возвращает или `String`, или такой же итератор). Напишите поверх этого итератора другой, уже «плоский».
12. Напишите итератор, который проходит по двум итератором.
13. Напишите метод, который получает на вход `Map<K, V>` и возвращает `Map`, где ключи и значения поменяны местами. Так как значения могут совпадать, то тип значения в `Map` будет уже не `K`, а `Collection<K>`:

## Map<V, Collection<K>>

14. Напишите программу, в которой создаются два потока, которые выводят на консоль своё имя по очереди.
15. Выведете состояние потока перед его запуском, после запуска и во время выполнения.
16. Создайте свою аннотацию Repeat с целочисленным параметром. Расширьте класс ThreadPoolExecutor и переопределите метод execute следующим образом: если экземпляр Runnable имеет аннотацию Repeat, то его метод run выполняется несколько раз (количество задается параметром в Repeat).
17. Требуется определить, сколько дорожных ограничений действовало в городе на определенную дату.  
Программа в качестве аргумент получает два параметра:  
Путь к файлу с данными  
Дата  
то есть она запускается следующим образом:  

```
java TrafficBlocks "PATH_TO_CSV_FILE" dd.MM.yyyy
```

  
Необходимо вывести количество действовавших ограничений движения транспорта на эту дату.
18. Напишите программу, которая с консоли считывает поисковый запрос, и выводит результат поиска по Википедии. Задача разбивается на 4 этапа:
  - a. Считать запрос
  - b. Сделать запрос к серверу
  - c. Распарсить ответ
  - d. Вывести результат
19. Напишите бота для Telegram, который будет:
  - a. Сообщать текущую погоду и прогноз на сутки в ответ на присланное местоположение.
  - b. Присылать прогноз погоды на сутки каждый день, если пользователь подписался на ежедневную рассылку. Подписка должна происходить с помощью команды /subscribe. Также должна быть предусмотрена возможность отписки (/unsubscribe).  
Для получения прогноза можно использовать <https://openweathermap.org/api>.

### 7.3.3. Перечень курсовых работ

- 1 Разработка клиент-серверного игрового приложения
- 2 Разработка музыкального плеера
- 3 Разработка игры в жанре «Платформер»
- 4 Разработка прикладной программы для контент-менеджеров
- 5 Разработка приложения для автоматизированного создания сайтов
- 6 Разработка настольной игры
- 7 Разработка приложения для обмена сообщениями с шифрованием
- 8 Разработка сервисного приложения с использованием API-интерфейсов
- 9 Разработка программы для автоматизации учета товаров на складе
- 10 Разработка программы-симулятора
- 11 Разработка автоматизированной системы управления персоналом
- 12 Разработка программы для решения систем уравнений
- 13 Разработка программы звукозаписи
- 14 Разработка системы управления складом предприятия
- 15 Разработка прикладного решения для поиска и удаления дубликатов файлов
- 16 Разработка приложения для поддержания здорового образа жизни
- 17 Разработка программы для конвертации валют
- 18 Разработка графического редактора
- 19 Разработка многопользовательской онлайн игры
- 20 Разработка программного комплекса учета приема-выдачи оборудования с использованием штрихового кодирования

### 7.4. Электронное портфолио обучающегося

В электронном портфолио обучающегося по дисциплине размещается

<http://portfolio.usue.ru>

- курсовая работа

Федеральное государственное бюджетное образовательное учреждение высшего образования

**УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ**

УТВЕРЖДЕНЫ

на заседании кафедры информационных  
технологий и статистики

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ**

**ТЕКУЩЕГО КОНТРОЛЯ**

**по дисциплине**

**Объектно-ориентированное программирование**

## Тест

### 1. Вопрос: Объектно-ориентированные технологии имеют следующие преимущества:

Ответ 1: Уменьшенную связность между модулями

Ответ 2: Повышенное качество кода

Ответ 3: Высокий уровень абстракции

Ответ 4: Все из вышеперечисленного

### 2. Вопрос: В Объектно-ориентированной Технологии слово "UML" означает

Ответ 1: Unified Module Language

Ответ 2: Unified Modeling Language

Ответ 3: Universal Module Leveling

Ответ 4: Universal Module Language

### 3. Вопрос: Объект имеет

Ответ 1: Поведение

Ответ 2: Состояние

Ответ 3: Атрибуты

Ответ 4: Все из вышеперечисленного

### 4. Вопрос: Какое из следующих утверждений верно?

Ответ 1: Объект - это экземпляр класса

Ответ 2: Класс - это абстрактное определение для множества объектов

Ответ 3: Объект может быть более чем в одном классе

Ответ 4: Объект имеет линию жизни

Ответ 5: Все из вышеперечисленного

### 5. Вопрос: Состояние объекта определяется

Ответ 1: Значением всех его атрибутов

Ответ 2: Его связями с другими объектами

Ответ 3: Его поведением в любой данный момент времени

Ответ 4: Операциями, которые он может выполнять

Ответ 5: Ответы 2 и 4

Ответ 6: Ответы 1 и 2

Ответ 7: Ответы 1, 3 и 4

Ответ 8: Ответы 1, 2 и 3

### 6. Вопрос: Структура класса

Ответ 1: Представлена в коде

Ответ 2: Представлена атрибутами и связями

Ответ 3: Представлена операциями

Ответ 4: Представлена взаимодействиями объектов

Ответ 5: Ответы 1 и 2

Ответ 6: Ответы 2 и 3

### 7. Вопрос: Какое из следующих утверждений правильное:

Ответ 1: Классы на диаграммах классов могут быть сгруппированы в пакеты, чтобы показать общую организацию модели (архитектуру системы)

Ответ 2: На диаграмме объектов наименования экземпляров должны быть выделены наклонным шрифтом

Ответ 3: Если пакет В зависит от пакета А, то любые изменения в А вызывают изменения в пакете В.

Ответ 4: Диаграммы объектов и диаграммы классов взаимозаменяемы

**8. Вопрос: Какое из следующих утверждений о кратности роли в ассоциации является неверным?**

Ответ 1: "1" - именно один и только один

Ответ 2: "0..n" - любое вещественное число, включая ноль

Ответ 3: "0..1" - ноль или один

Ответ 4: "3..7" - 3 или 7

Ответ 5: "3, 7" - 3 или 7

**9. Вопрос: На диаграмме требований (Use Case diagram) есть три основных элемента, это:**

Ответ 1: Объекты, действия и связи

Ответ 2: Субъекты (акторы), сообщения и связи

Ответ 3: Объекты, требования и связи

Ответ 4: Субъекты (акторы), требования и связи

**10. Вопрос: Какое утверждение описывает отношение наследования наилучшим образом?**

Ответ 1: Отношение "имеет"

Ответ 2: Отношение "является"

Ответ 3: Отношение "выполняет"

Ответ 4: Отношение "использует"

Ответ 5: Отношение "является элементом"

**11. Вопрос: Какое утверждение описывает отношение агрегации наилучшим образом?**

Ответ 1: Отношение "имеет"

Ответ 2: Отношение "является"

Ответ 3: Отношение "выполняет"

Ответ 4: Отношение "использует"

Ответ 5: Отношение "является частью"

**12. Вопрос: Операция в классе «нечто» вызывает операцию в классе «что-то». Других отношений между двумя классами нет. Каким типом отношений является данный?**

Ответ 1: Ассоциация

Ответ 2: Агрегация

Ответ 3: Наследование

Ответ 4: Реализация

Ответ 5: Зависимость

**13. Вопрос: Машина имеет 4 колеса. Какой тип отношений имеют класс "Машина" и класс "Колесо"?**

Ответ 1: Ассоциация

Ответ 2: Агрегация

Ответ 3: Наследование

Ответ 4: Реализация

**14. Вопрос: Студент посещает несколько аудиторий. Аудитория может вмещать несколько студентов. Какой тип отношений существует между студентом и аудиторией?**

Ответ 1: Ассоциация

Ответ 2: Агрегация

Ответ 3: Наследование

Ответ 4: Реализация

Ответ 5: Зависимость

**15. Вопрос: Какое из следующих утверждений правильное:**

Ответ 1: Классы на диаграммах классов могут быть сгруппированы в пакеты, чтобы показать общую организацию модели (архитектуру системы)

Ответ 2: На диаграмме объектов наименования экземпляров должны быть выделены наклонным шрифтом

Ответ 3: Если пакет В зависит от пакета А, то любые изменения в А вызывают изменения в пакете В.

Ответ 4: Диаграммы объектов и диаграммы классов взаимозаменяемы

**17. Вопрос: Выберите отношение, которое может иметь кратность, отличную от “1”.**

Ответ 1: Ассоциация

Ответ 2: Агрегация

Ответ 3: Наследование

Ответ 4: Реализация

Ответ 5: Зависимость

Ответ 6: Все из вышеперечисленного

Ответ 7: Ответы 1, 3 и 4

Ответ 8: Ответы 1 и 2

Ответ 9: Ответы 3, 4 и 5

**18. Вопрос: Какое из следующих утверждений является неверным?**

Ответ 1: Класс может иметь отношение к самому себе.

Ответ 2: Объект может иметь отношение к другим объектам того же класса.

Ответ 3: Класс может иметь только одно отношение к другому классу.

Ответ 4: Отношение может существовать с мощностью “0”.

Ответ 5: Класс может существовать без каких-либо отношений к другим объектам.

**19. Вопрос: Какой тип отношений к самому себе может иметь класс?**

Ответ 1: Ассоциация

Ответ 2: Агрегация

Ответ 3: Наследование

Ответ 4: Реализация

Ответ 5: Зависимость

Ответ 6: Ответы 1 и 2

Ответ 7: Ответы 3 и 5

Ответ 8: Ответы 4 и 5

Ответ 9: Ничего из вышеперечисленного

**20. Вопрос: Что такое “актор” в анализе варианта использования?**

Ответ 1: Любая сущность, которая взаимодействует с системой.

Ответ 2: Термин для бизнес-объектов, которые присутствуют в системе.

Ответ 3: То, что реагирует на внешние воздействия системы.

Ответ 4: В анализе варианта использования этого термина нет.

**21. Вопрос: Диаграмма Последовательности содержит**

Ответ 1: Объекты

Ответ 2: Сообщения

Ответ 3: Видимость объекта

Ответ 4: Временные ограничения

Ответ 6: Все из вышеперечисленного

Ответ 7: Все из вышеперечисленного за исключением 3

Ответ 9: Все из вышеперечисленного за исключением 1

**22. Вопрос: Диаграмма Состояний содержит**

- Ответ 1: Состояния класса
- Ответ 2: Переходы между состояниями
- Ответ 3: Действия, выполняемые классом
- Ответ 4: События, которые являются действиями в классе
- Ответ 5: Все из вышеперечисленного

**23. Вопрос: Диаграмма Состояний может иметь только одно (один)**

- Ответ 1: Конечное состояние
- Ответ 2: Начальное состояние
- Ответ 3: Действие в состоянии
- Ответ 4: Переход из состояния
- Ответ 5: Все из вышеперечисленного

**24. Вопрос: Какое из следующих утверждений является неверным?**

- Ответ1: В какой-либо момент времени объект может быть более чем в одном состоянии
- Ответ2: Автомат может иметь множество конечных состояний
- Ответ3: Автомат может запоминать, какое подсостояние было последним состоянием
- Ответ4: Переход может срабатывать условно

**25. Вопрос: Какое из следующих утверждений о диаграмме состояний является правильным:**

- Ответ 1: Все действия на диаграмме состояний связаны с переходами
- Ответ 2: Событие может явиться причиной того, что объект останется в том же состоянии, предшествующем событию
- Ответ 3: Если объект выходит из состояния, он не может вернуться в это состояние.
- Ответ 4: Два различных перехода из одного состояния могут перекрывать друг друга (они могут быть вызваны одним и тем же событием)

**26. Что означает аббревиатура RUP?**

- Ответ 1: Rational Unified Process
- Ответ 2: Rational Universal Process
- Ответ 3: Rational Unified Program

**28. Для чего нужен словарь модели**

- Ответ1: Чтобы ограничивать количество используемых терминов
- Ответ2: Чтобы формировать терминологию предметной области
- Ответ3: Чтобы добавлять новые термины

**29. Названия прецедентов должны начинаться:**

- Ответ1: с глагола или существительного
- Ответ2: с глагола
- Ответ3: с прилагательного
- Ответ4: с глагола или прилагательного



## Практическая работа №1

### Вариант 1

1. Описать класс, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне. Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и свойство, позволяющее получить его текущее состояние. При выходе за границы диапазона выбрасываются исключения

Написать программу, демонстрирующую все разработанные элементы класса.

2. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, вычисления площади, периметра и точки пересечения медиан.

Описать свойства для получения состояния объекта. При невозможности построения треугольника выбрасывается исключение.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (по автору, по году издания или категории), добавления книг в библиотеку, удаления книг из нее, доступа к книге по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

4. Описать класс «поезд», содержащий следующие закрытые поля:

- название пункта назначения;
- номер поезда (может содержать буквы и цифры);
- время отправления.

Предусмотреть свойства для получения состояния объекта.

Описать класс «вокзал», содержащий закрытый массив поездов. Обеспечить следующие возможности:

- вывод информации о поезде по номеру с помощью индекса;
- вывод информации о поездах, отправляющихся после введенного с клавиатуры времени;
- перегруженную операцию сравнения, выполняющую сравнение времени отправления двух поездов;
- вывод информации о поездах, отправляющихся в заданный пункт назначения.

Информация должна быть отсортирована по времени отправления. Написать программу, демонстрирующую все разработанные элементы классов.

### Вариант 2

1. Описать класс, реализующий шестнадцатеричный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне. Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и

свойство, позволяющее получить его текущее состояние. При выходе за границы диапазона выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

2. Описать класс, представляющий круг. Предусмотреть методы для создания объектов, вычисления площади круга, длины окружности и проверки попадания заданной точки внутрь круга. Описать свойства для получения состояния объекта.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс «записная книжка». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по фамилии и доступа к записи по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

4. Описать класс «поезд», содержащий следующие закрытые поля:

- название пункта назначения;
- номер поезда (может содержать буквы и цифры);
- время отправления.

Предусмотреть свойства для получения состояния объекта.

Описать класс «вокзал», содержащий закрытый массив поездов. Обеспечить следующие возможности:

- вывод информации о поезде по номеру с помощью индекса;
- вывод информации о поездах, отправляющихся после введенного с клавиатуры времени;
- перегруженную операцию сравнения, выполняющую сравнение времени отправления двух поездов;
- вывод информации о поездах, отправляющихся в заданный пункт назначения.

Информация должна быть отсортирована по времени отправления. Написать программу, демонстрирующую все разработанные элементы классов.

### *Вариант 3*

1. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Описать свойства для получения состояния объекта. При невозможности построения треугольника выбрасывается исключение.

Написать программу, демонстрирующую все разработанные элементы класса

2. Описать класс для работы со строкой, позволяющей хранить только двоичное число и выполнять с ним арифметические операции. Предусмотреть инициализацию с проверкой допустимости значений. В случае недопустимых значений выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, имени, дате рождения), добавления и удаления записей, сортировки по разным полям, доступа к записи по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

4. Описать класс «товар», содержащий следующие закрытые поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях.

Предусмотреть свойства для получения состояния объекта.

Описать класс «склад», содержащий закрытый массив товаров. Обеспечить следующие возможности:

- вывод информации о товаре по номеру с помощью индекса;
- вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, выдать соответствующее сообщение;
- сортировку товаров по названию магазина, по наименованию и по цене;
- перегруженную операцию сложения товаров, выполняющую сложение их цен.

Написать программу, демонстрирующую все разработанные элементы классов.

#### *Вариант 4*

1. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность отдельного изменения составных частей адреса и проверки допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

2. Описать класс дробей — рациональных чисел, являющихся отношением двух целых чисел.

Предусмотреть методы сложения, вычитания, умножения и деления дробей.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс «автостоянка» для хранения сведений об автомобилях. Для каждого автомобиля записываются госномер, цвет, фамилия владельца и признак присутствия на стоянке. Обеспечить возможность поиска автомобиля по разным критериям, вывода списка присутствующих и отсутствующих на стоянке автомобилей, доступа к имеющимся сведениям по номеру места.

Написать программу, демонстрирующую все разработанные элементы класса.

4. Описать класс «товар», содержащий следующие закрытые поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях.

Предусмотреть свойства для получения состояния объекта.

Описать класс «склад», содержащий закрытый массив товаров. Обеспечить следующие возможности:

- вывод информации о товаре по номеру с помощью индекса;

- вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, выдать соответствующее сообщение;
- сортировку товаров по названию магазина, по наименованию и по цене;
- перегруженную операцию сложения товаров, выполняющую сложение их цен.

Написать программу, демонстрирующую все разработанные элементы классов.

### *Вариант 5*

1. Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.

Написать программу, демонстрирующую все разработанные элементы класса.

2. Описать класс «файл», содержащий сведения об имени, дате создания и длине файла. Предусмотреть инициализацию с проверкой допустимости значений полей. В случае недопустимых значений полей выбрасываются исключения. Описать метод добавления информации в конец файла и свойства для получения состояния файла.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс «колода карт», включающий закрытый массив элементов класса «карта». В карте хранятся масть и номер. Обеспечить возможность вывода карты по номеру, вывода всех карт, перемешивания колоды и выдачи всех карт из колоды поодиночке и по 6 штук в случайном порядке.

Написать программу, демонстрирующую все разработанные элементы классов.

4. Описать класс «самолет», содержащий следующие закрытые поля:

- название пункта назначения;
- шестизначный номер рейса;
- время отправления.

Предусмотреть свойства для получения состояния объекта.

Описать класс «аэропорт», содержащий закрытый массив самолетов. Обеспечить

следующие возможности:

- вывод информации о самолете по номеру рейса с помощью индекса;
- вывод информации о самолетах, отправляющихся в течение часа после введенного с клавиатуры времени;
- вывод информации о самолетах, отправляющихся в заданный пункт назначения;
- перегруженную операцию сравнения, выполняющую сравнение времени отправления двух самолетов.

Информация должна быть отсортирована по времени отправления.

Написать программу, демонстрирующую все разработанные элементы классов.



## Практическая работа №2

### Вариант 1

Написать Windows-приложение, которое выполняет анимацию изображения.

Создать меню с командами **Show picture**, **Choose**, **Animate**, **Stop**, **Quit**.

Команда **Quit** завершает работу приложения. При выборе команды **Show picture** в центре экрана рисуется объект, состоящий из нескольких графических примитивов.

При выборе команды **Choose** открывается диалоговое окно, содержащее:

- поле типа **TextBox** с меткой **Speed** для ввода скорости движения объекта;
- группу **Direction** из двух переключателей (**Up-Down**, **Left-Right**) типа **RadioButton** для выбора направления движения;
- кнопку типа **Button**.

По команде **Animate** объект начинает перемещаться в выбранном направлении до края окна и обратно с заданной скоростью, по команде **Stop** — прекращает движение.

### Вариант 2

Написать Windows-приложение, которое по заданным в файле исходным данным строит график или столбиковую диаграмму.

Создать меню с командами **Input data**, **Choose**, **Line**, **Bar**, **Quit**.

Команды **Line** и **Bar** недоступны. Команда **Quit** завершает работу приложения.

При выборе команды **Input data** из файла читаются исходные данные (файл сформировать самостоятельно).

По команде **Choose** открывается диалоговое окно, содержащее:

- список для выбора цвета графика типа **ListBox**;
- группу из двух переключателей (**Line**, **Bar**) типа **RadioButton**;
- кнопку типа **Button**.

Обеспечить возможность ввода цвета и выбора режима: построение графика (**Line**) или столбиковой диаграммы (**Bar**). После указания параметров становится доступной соответствующая команда меню.

По команде **Line** или **Bar** в главном окне приложения выбранным цветом строится график или диаграмма. Окно должно содержать заголовок графика или диаграммы, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

### Вариант 3

Написать Windows-приложение, которое строит графики четырех заданных функций.

Создать меню с командами **Chart**, **Build**, **Clear**, **About**, **Quit**.

Команда **Quit** завершает работу приложения. При выборе команды **About** открывается окно с информацией о разработчике.

Команда **Chart** открывает диалоговое окно, содержащее:

- список для выбора цвета графика типа **ListBox**;
- список для выбора типа графика типа **ListBox**, содержащий четыре пункта: **sin(x)**, **sin(x+π/4)**, **cos(x)**, **cos(x-π/4)**;
- кнопку типа **Button**.

Обеспечить возможность выбора цвета и вида графика. После щелчка на кнопке **OK** в главном окне приложения строится график выбранной функции на интервале от  $-\pi/2$  до  $+\pi/2$ . Окно должно содержать заголовок графика, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Команда **Clear** очищает окно.

#### *Вариант 4*

Написать Windows-приложение — графическую иллюстрацию сортировки методом выбора.

Создать меню с командами **File**, **Animate**, **About**, **Exit**.

Команда **Animate** недоступна. Команда **Exit** завершает работу приложения. Команда **About** открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда **File**) использовать объект класса `OpenFileDialog`.

Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа).

После чтения данных становится доступной команда **Animate**.

При выборе команды **Animate** в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовок. Изображение должно занимать все окно.

#### *Вариант 5*

Написать Windows-приложение — графическую иллюстрацию сортировки методом пузырька.

Создать меню с командами **File**, **Animate**, **About**, **Exit**.

Команда **Animate** недоступна. Команда **Exit** завершает работу приложения. Команда **About** открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда **File**) использовать объект класса `OpenFileDialog`.

Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа).

После чтения данных становится доступной команда **Animate**.

При выборе команды **Animate** в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовок. Изображение должно занимать все окно.

Федеральное государственное бюджетное образовательное учреждение высшего образования

**УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ**

УТВЕРЖДЕНЫ

на заседании кафедры информационных  
технологий и статистики

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ**

**ПРОМЕЖУТОЧНОГО КОНТРОЛЯ**

**по дисциплине**

**Объектно-ориентированное программирование**



## ЗАЧЕТ

### Экзаменационный билет №1

1. Базовые типы данных. Примитивные типы, ссылочные типы.
2. Массивы. Инициализация и индексация массива, многомерные массивы.
3. Заполните массив случайными числами и выведите максимальное, минимальное и среднее значение. Для генерации случайного числа используйте метод `Math.random()`, который возвращает значение в промежутке `[0, 1]`.

### Экзаменационный билет №2

1. Логические операторы. Приоритет выполнения. Сокращенные версии логических операторов `&&`, `||` и их отличия от `|`, `&`.
2. Понятие класса и объекта. Определение понятий. Статический и не статический контекст класса.
3. Реализуйте алгоритм сортировки пузырьком для сортировки массива.

### Экзаменационный билет №3

1. Арифметические операторы. Операторы инкремента, декремента, приоритет операций.
2. Понятие класса и объекта. Определение понятий. Статический и не статический контекст класса.
3. Напишите программу, которая выводит на консоль простые числа в промежутке от `[2, 100]`. Используйте для решения этой задачи оператор `%` (остаток от деления) и циклы.

### Экзаменационный билет №4

1. Приведение типов (явное и автоматическое). Константы и переменные.
2. Понятие класса и объекта. Члены класса. Методы, поля, конструкторы, блоки инициализации. Ключевые слова `abstract` и `final`.
3. Дан массив целых чисел и ещё одно целое число. Удалите все вхождения этого числа из массива (пропусков быть не должно).

### Экзаменационный билет №5

1. Операторы ветвления. Условный оператор. Тернарный условный оператор.
2. Понятие класса и объекта. Внутренние классы, вложенные классы, анонимные классы, определение, особенности.
3. 5. Создайте класс, который описывает вектор (в трёхмерном пространстве).

У него должны быть:

- конструктор с параметрами в виде списка координат `x, y, z`

- метод, вычисляющий длину вектора. Корень можно посчитать с помощью `Math.sqrt()`:

`Math.sqrt(x2+y2+z2)`

- метод, вычисляющий скалярное произведение:

`x1x2+y1y2+z1z2`

- метод, вычисляющий векторное произведение с другим вектором:

`(y1z2-z1y2, z1x2-x1z2, x1y2-y1x2)`

- метод, вычисляющий угол между векторами (или косинус угла): косинус угла между векторами равен скалярному произведению векторов, деленному на произведение модулей (длин) векторов:

`(a, b)/|a|·|b|`

- методы для суммы и разности:

`(x1+x2, y1+y2, z1+z2)`

`(x1-x2, y1-y2, z1-z2)`

- статический метод, который принимает целое число N, и возвращает массив случайных векторов размером N. Если метод возвращает вектор, то он должен возвращать новый объект, а не менять базовый. То есть, нужно реализовать шаблон "Неизменяемый объект"

#### Экзаменационный билет №6

1. Операторы ветвления. Оператор выбора. Его отличия от условного оператора.
2. Основопологающие принципы ООП. Инкапсуляция. Средства реализации инкапсуляции. Модификаторы доступа.
3. Напишите класс, конструктор которого принимает два массива: массив значений и массив весов значений. Класс должен содержать метод, который будет возвращать элемент из первого массива случайным образом, с учётом его веса. Пример: Дан массив [1, 2, 3], и массив весов [1, 2, 10]. В среднем, значение «1» должно возвращаться в 2 раза реже, чем значение «2» и в десять раз реже, чем значение «3».

#### Экзаменационный билет №7

1. Циклы. Операторы цикла и их отличия.
2. Основопологающие принципы ООП. Наследование. Управление наследованием.
3. 7. Напишите метод, который проверяет, входит ли в массив заданный элемент или нет. Используйте перебор и двоичный поиск для решения этой задачи. Сравните время выполнения обоих решений для больших массивов (например, 100000000 элементов).

#### Экзаменационный билет №8

1. Массивы. Инициализация и индексация массива, многомерные массивы.

2. Основополагающие принципы ООП. Полиморфизм. Средства реализации полиморфизма.

3. Найдите корень уравнения

$$\cos(x^5) + x^4 - 345.3 * x - 23 = 0$$

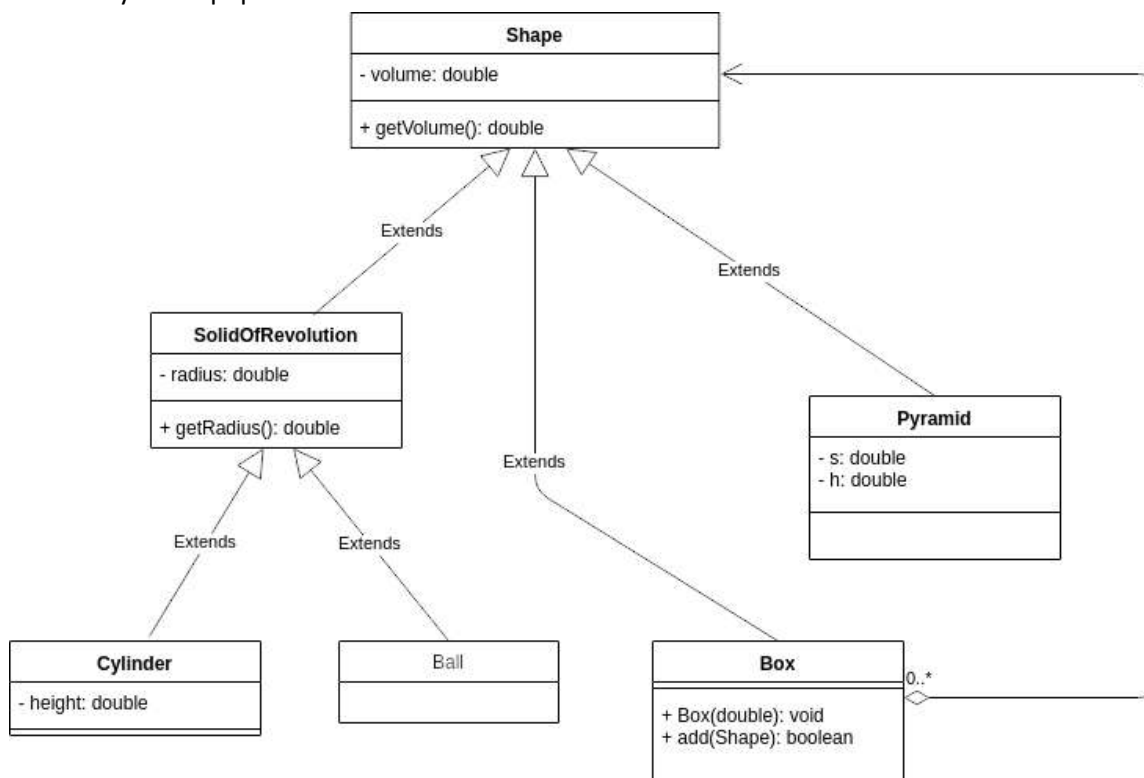
на отрезке [0; 10] с точностью по x не хуже, чем 0.001. Известно, что на этом промежутке корень единственный. Используйте для этого метод деления отрезка пополам (и рекурсию).

### Экзаменационный билет №9

1. Понятие класса и объекта. Определение понятий. Статический и не статический контекст класса.

2. Сигнатура метода. Понятия перегрузки и переопределения..

3 Реализуйте иерархию классов:



Класс Box

является контейнером, он можем содержать в себе другие фигуры. Метод add() принимает на вход Shape. Нужно добавлять новые фигуры до тех пор, пока для них хватает места в Box (будем считать только объём, игнорируя форму. Допустим, мы переливаем жидкость). Если места для добавления новой фигуры не хватает, то метод должен вернуть false..

### Экзаменационный билет №10

1. Понятие класса и объекта. Члены класса. Методы, поля, конструкторы, блоки инициализации. Ключевые слова abstract и final.

2. Базовые типы данных. Примитивные типы, ссылочные типы.

3. Постройте частотный словарь букв русского (или английского) алфавита. Опустим проблему выбора и анализа корпуса языка, достаточно будет взять текст небольшой длины).

### Экзаменационный билет №11

1. Понятие класса и объекта. Внутренние классы, вложенные классы, анонимные классы, определение, особенности..
2. Логические операторы. Приоритет выполнения. Сокращенные версии логических операторов &&, || и их отличия от |, &.
3. Заполните массив случайным числами и выведите максимальное, минимальное и среднее значение. Для генерации случайного числа используйте метод `Math.random()`, который возвращает значение в промежутке `[0, 1]`.

### Экзаменационный билет №12

1. Основопологающие принципы ООП. Инкапсуляция. Средства реализации инкапсуляции. Модификаторы доступа.
2. Основопологающие принципы ООП. Инкапсуляция. Средства реализации инкапсуляции. Модификаторы доступа.
3. Реализуйте алгоритм сортировки пузырьком для сортировки массива.

### Экзаменационный билет №13

1. Основопологающие принципы ООП. Наследование. Управление наследованием.
2. Приведение типов (явное и автоматическое). Константы и переменные.
3. Напишите программу, которая выводит на консоль простые числа в промежутке от `[2, 100]`. Используйте для решения этой задачи оператор `"%"` (остаток от деления) и циклы.

### Экзаменационный билет №14

1. Основопологающие принципы ООП. Полиморфизм. Средства реализации полиморфизма.
2. Операторы ветвления. Условный оператор. Тернарный условный оператор.
3. Дан массив целых чисел и ещё одно целое число. Удалите все вхождения этого числа из массива (пропусков быть не должно).

### Экзаменационный билет №15

1. Сигнатура метода. Понятия перегрузки и переопределения.
2. Операторы ветвления. Оператор выбора. Его отличия от условного оператора.
3. 5. Создайте класс, который описывает вектор (в трёхмерном пространстве).

У него должны быть:

- конструктор с параметрами в виде списка координат `x, y, z`

- метод, вычисляющий длину вектора. Корень можно посчитать с помощью `Math.sqrt()`:

`Math.sqrt(x2+y2+z2)`

- метод, вычисляющий скалярное произведение:

`x1x2+y1y2+z1z2`

- метод, вычисляющий векторное произведение с другим вектором:

`(y1z2-z1y2, z1x2-x1z2, x1y2-y1x2)`

- метод, вычисляющий угол между векторами (или косинус угла): косинус угла между векторами равен скалярному произведению векторов, деленному на произведение модулей (длин) векторов:

`(a, b)/|a|·|b|`

- методы для суммы и разности:

`(x1+x2, y1+y2, z1+z2)`

`(x1-x2, y1-y2, z1-z2)`

- статический метод, который принимает целое число N, и возвращает массив случайных векторов размером N. Если метод возвращает вектор, то он должен возвращать новый объект, а не менять базовый. То есть, нужно реализовать шаблон "Неизменяемый объект".

#### Экзаменационный билет №16

1. Арифметические операторы. Операторы инкремента, декремента, приоритет операций.
2. Основопологающие принципы ООП. Инкапсуляция. Средства реализации инкапсуляции. Модификаторы доступа.
3. Напишите класс, конструктор которого принимает два массива: массив значений и массив весов значений. Класс должен содержать метод, который будет возвращать элемент из первого массива случайным образом, с учётом его веса. Пример: Дан массив [1, 2, 3], и массив весов [1, 2, 10]. В среднем, значение «1» должно возвращаться в 2 раза реже, чем значение «2» и в десять раз реже, чем значение «3».

#### Экзаменационный билет №17

1. Приведение типов (явное и автоматическое). Константы и переменные.
2. Основопологающие принципы ООП. Наследование. Управление наследованием.
3. Напишите метод, который проверяет, входит ли в массив заданный элемент или нет. Используйте перебор и двоичный поиск для решения этой задачи. Сравните время выполнения обоих решений для больших массивов (например, 100000000 элементов).

#### Экзаменационный билет №18

1. Операторы ветвления. Условный оператор. Тернарный условный оператор.

2. Основополагающие принципы ООП. Полиморфизм. Средства реализации полиморфизма.

3. Найдите корень уравнения

$$\cos(x^5) + x^4 - 345.3 * x - 23 = 0$$

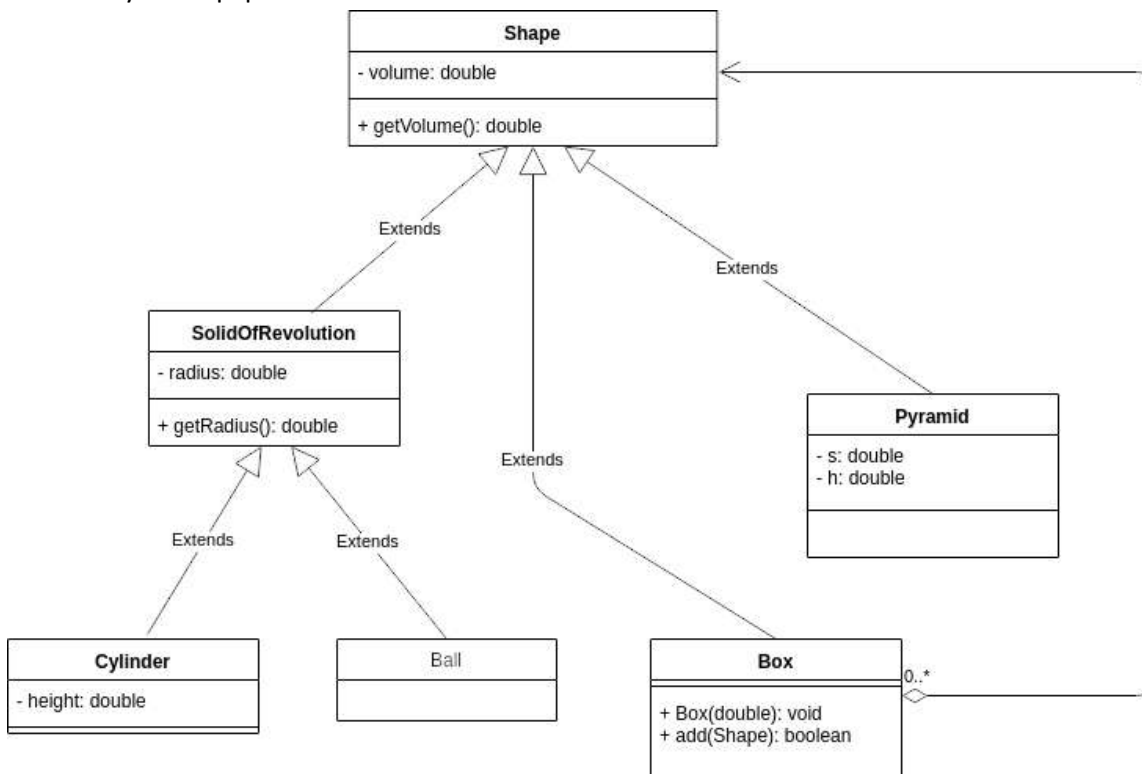
на отрезке  $[0; 10]$  с точностью по  $x$  не хуже, чем  $0.001$ . Известно, что на этом промежутке корень единственный. Используйте для этого метод деления отрезка пополам (и рекурсию).

### Экзаменационный билет №19

1. Понятие класса и объекта. Определение понятий. Статический и не статический контекст класса.

2. Сигнатура метода. Понятия перегрузки и переопределения.

3. Реализуйте иерархию классов:



Класс **Box** является контейнером, он может содержать в себе другие фигуры. Метод *add()* принимает на вход *Shape*. Нужно добавлять новые фигуры до тех пор, пока для них хватает места в *Box* (будем считать только объём, игнорируя форму. Допустим, мы переливаем жидкость). Если места для добавления новой фигуры не хватает, то метод должен вернуть **false**.

### Экзаменационный билет №20

1. Базовые типы данных. Примитивные типы, ссылочные типы.

2. Приведение типов (явное и автоматическое). Константы и переменные.

3. Постройте частотный словарь букв русского (или английского) алфавита. Опустим проблему выбора и анализа корпуса языка, достаточно будет взять текст небольшой длины)

## ЭКЗАМЕН

### Экзаменационный билет №1

1. Базовые типы данных. Примитивные типы, ссылочные типы.
2. Исключения. Определение исключения, иерархия исключений
3. Напишите класс BaseConverter для конвертации из градусов по Цельсию в Кельвины, Фаренгейты, и так далее. У метода должен быть метод convert, который и делает конвертацию.

### Экзаменационный билет №2

1. Логические операторы. Приоритет выполнения. Сокращенные версии логических операторов &&, || и их отличия от |, &.
2. Исключения. Проверяемые исключения, определение, особенности, пример реализации собственных исключений.
3. Напишите свой класс StringBuilder с поддержкой операции undo. Для этого делегируйте все методы стандартному StringBuilder, а в собственном классе храните список всех операций для выполнения undo(). Это будет реализацией шаблона «Команда».

### Экзаменационный билет №3

1. Арифметические операторы. Операторы инкремента, декремента, приоритет операций.
2. Исключения. Непроверяемые исключения, определение, особенности, пример реализации собственных исключений.
3. Напишите свой класс StringBuilder, с возможностью оповещения других объектов об изменении своего состояния. Для этого делегируйте все методы стандартному StringBuilder, а в собственном классе реализуйте шаблон проектирования «Наблюдатель».

### Экзаменационный билет №4

1. Приведение типов (явное и автоматическое). Константы и переменные.
2. Исключения. Error. Определение, особенности, примеры.
3. Напишите метод filter, который принимает на вход массив (любого типа) и реализацию интерфейса Filter с методом apply(Object o), чтобы убрать из массива лишнее. Проверьте как он работает на строках или других объектах

#### Экзаменационный билет №5

1. Операторы ветвления. Условный оператор. Тернарный условный оператор.
2. Исключения. Обработка исключений. Исключения и наследование.
3. Напишите метод fill, который принимает массив объектов, и реализацию интерфейса Function (или своего). Метод fill должен заполнить массив, получая новое значение по индексу с помощью реализации интерфейса Function.

#### Экзаменационный билет №6

1. Операторы ветвления. Оператор выбора. Его отличия от условного оператора.
2. Потоки ввода-вывода. Основные типы потоков ввода-вывода, основные методы, различия
3. Постройте частотный словарь слов русского (или английского) алфавита. Опустим проблему выбора и анализа корпуса языка, достаточно будет взять текст небольшой длины).

#### Экзаменационный билет №7

1. Циклы. Операторы цикла и их отличия.
2. Символьные потоки ввода-вывода. Основные представители, описание.
3. Напишите метод, который на вход получает коллекцию объектов, а возвращает коллекцию уже без дубликатов.

#### Экзаменационный билет №8

1. Массивы. Инициализация и индексация массива, многомерные массивы
2. Байтовые потоки ввода/вывода. Основные представители, описание.
3. Напишите метод, который добавляет 1000000 элементов в ArrayList и LinkedList. Напишите еще один метод, который выбирает из заполненного списка элемент наугад 100000 раз. Замерьте время, которое потрачено на это. Сравните результаты и предположите, почему они именно такие.

#### Экзаменационный билет №9

1. Понятие класса и объекта. Определение понятий. Статический и не статический контекст класса.
2. Сетевое взаимодействие с помощью Socket. Описание, механизм работы, примеры использования.
3. Написать итератор по массиву.

#### Экзаменационный билет №10



1. Понятие класса и объекта. Члены класса. Методы, поля, конструкторы, блоки инициализации. Ключевые слова `abstract` и `final`
2. Сериализация и десериализация объектов. Определение, условия, реализация, внедрение в процесс сериализации.
3. Написать итератор по двумерному массиву.

#### Экзаменационный билет №11

1. Понятие класса и объекта. Внутренние классы, вложенные классы, анонимные классы, определение, особенности.
2. Методы класса `Object` `hashCode()` и `equals()`. Описание, правила реализации.
3. Дан итератор. Метод `next()` возвращает либо `String`, либо итератор такой же структуры (то есть который опять возвращает или `String`, или такой же итератор). Напишите поверх этого итератора другой, уже «плоский».

#### Экзаменационный билет №12

1. Основопологающие принципы ООП. Инкапсуляция. Средства реализации инкапсуляции. Модификаторы доступа.
2. Интерфейсы `Comparator` и `Comparable`. Описание, различия, примеры использования.
3. Напишите итератор, который проходит по двум итератором.

#### Экзаменационный билет №13

1. Основопологающие принципы ООП. Наследование. Управление наследованием.
2. `Collection framework`. Коллекции типа `Set`. Описание, представители. Механизм работы, различия реализаций.
3. Напишите метод, который получает на вход `Map<K, V>` и возвращает `Map`, где ключи и значения поменяны местами. Так как значения могут совпадать, то тип значения в `Map` будет уже не `K`, а `Collection<K>`:

`Map<V, Collection<K>>`

#### Экзаменационный билет №14

1. 14. Основопологающие принципы ООП. Полиморфизм. Средства реализации полиморфизма.
2. `Collection framework`. Коллекции типа `List`. Описание, представители. Механизм работы, различия реализаций.

3. Напишите программу, в которой создаются два потока, которые выводят на консоль своё имя по очереди.

#### Экзаменационный билет №15

1. Сигнатура метода. Понятия перегрузки и переопределения.
2. Collection framework. Коллекции типа Map. Описание, представители. Механизм работы, различия реализаций.
3. Выведете состояние потока перед его запуском, после запуска и во время выполнения.

#### Экзаменационный билет №16

1. Понятие класса и интерфейса: абстрактные классы, абстрактные методы. Отличие абстрактного класса от интерфейса.
2. Collection framework. Коллекции типа Queue/Deque. Описание, представители. Механизм работы, различия.
3. Создайте свою аннотацию Repeat с целочисленным параметром. Расширьте класс ThreadPoolExecutor и переопределите метод execute следующим образом: если экземпляр Runnable имеет аннотацию Repeat, то его метод run выполняется несколько раз (количество задается параметром в Repeat).

#### Экзаменационный билет №17

1. Интерфейсы: определение, реализация, наследование.
2. Потoki исполнения. Понятия потока и процесса. Жизненный цикл потока
3. Требуется определить, сколько дорожных ограничений действовало в городе на определенную дату.

Программа в качестве аргумент получает два параметра:

Путь к файлу с данными

Дата

то есть она запускается следующим образом:

```
java TrafficBlocks "PATH_TO_CSV_FILE" dd.MM.yyyy
```

Необходимо вывести количество действовавших ограничений движения транспорта на эту дату.

#### Экзаменационный билет №18

1. Перечисления (Enum): определение, реализация, особенности.
2. Потoki исполнения. Способы создания потоков исполнения. Класс Thread и его методы start(), run(), sleep(), yield().

3. Напишите программу, которая с консоли считывает поисковый запрос, и выводит результат поиска по Википедии. Задача разбивается на 4 этапа:

- a. Считать запрос
- b. Сделать запрос к серверу
- c. Распарсить ответ
- d. Вывести результат

#### Экзаменационный билет №19

1. Дженирики: Определение, реализация, примеры.

2. Потоки исполнения. Прерывание потоков.

3. Напишите бота для Telegram, который будет:

- a. Сообщать текущую погоду и прогноз на сутки в ответ на присланное местоположение.
- b. Присылать прогноз погоды на сутки каждый день, если пользователь подписался на ежедневную рассылку. Подписка должна происходить с помощью команды `/subscribe`. Также должна быть предусмотрена возможность отписки (`/unsubscribe`).

Для получения прогноза можно использовать <https://openweathermap.org/api>.

#### Экзаменационный билет №20

1. Дженирики: параметризованные методы, маски (wildcards). Примеры.

2. Потоки исполнения. Взаимодействие потоков. Понятия Мьютекс, Монитор, Критическая секция и их реализация в языке Java.

3. Напишите класс `BaseConverter` для конвертации из градусов по Цельсию в Кельвины, Фаренгейты, и так далее. У метода должен быть метод `convert`, который и делает конвертацию.

## Методические рекомендации по выполнению курсовой работы по дисциплине «Объектно-ориентированное программирование»

### Общие положения

Осуществление будущей профессиональной деятельности бакалавра по направлениям «Информатика и вычислительная техника», «Прикладная информатика», «Математическое обеспечение и администрирование информационных систем» предполагает использование полученных в процессе обучения систематизированных теоретических знаний для решения практических задач. Важнейшим фактором формирования высокого профессионального уровня специалиста, его умений и навыков является самостоятельная работа, выполняемая в форме курсовых работ по изучаемым дисциплинам.

В процессе обучения студенты указанных выше направлений подготовки бакалавриата должны выполнить курсовую работу по дисциплине «Объектно-ориентированное программирование».

Задачи, решаемые в процессе выполнения курсовой работы:

- освоение методологии алгоритмизации и объектно-ориентированного программирования;
- уяснение принципов разработки программного обеспечения;
- ознакомление с методами и средствами объектно-ориентированного программирования, находящимся за пределами учебного курса.

Курсовая работа должна быть сдана научному руководителю не позднее, чем за 2 недели до сессии (или по специальной договоренности с руководителем при форс-мажорных обстоятельствах) с целью ее своевременной проверки и защиты. Студент, не защитивший курсовую работу, не допускается к сдаче экзамена по данной дисциплине.

## Выбор темы и консультирование

При определении темы курсовой работы необходимо учитывать возможность использования современных методов, алгоритмов и соответствующего программного обеспечения.

Тематика работ определяется кафедрой в соответствии с учебным планом, направлениями научно-исследовательской работы студента.

В течение семестра проводятся консультации с руководителем курсовой работы, во время которой обсуждаются сложности, с которыми сталкивается студент при выполнении задания. Руководитель оказывает помощь студенту в составлении графика работы, подборе необходимой литературы, проводит консультации, проверяет выполнение проекта и организует его защиту. Однако контроль руководителя не освобождает студента от ответственности за правильность и своевременность выполнения проекта. Основной объем при выполнении курсового проекта составляет самостоятельная работа.

Защита курсовой работы проводится в следующем порядке:

а) не позднее, чем за две недели до защиты курсовой работы студентом руководитель рассматривает код написанной программы (или демонстрационные примеры использования выбранной технологии) и допускает до защиты;

б) студент оформляет курсовую работу и готовит доклад для выступления;

в) защита выполняется публично, путем доклада перед комиссией, в которой обязательно присутствует преподаватель дисциплины, а также другие преподаватели кафедры информационных технологий и статистики и практикующие специалисты в области разработки программного обеспечения. В результате защиты оценивается сложность работы, качество оформления материалов, выполненный доклад. На основании усреднения баллов членов комиссии выставляется оценка, даются рекомендации по публикации результатов, размещении результатов проекта на сайте кафедры.

Студенты могут по своему усмотрению выбрать темы, которые не включены в прилагаемый список, согласовав предлагаемые оригинальные варианты с руководителем. Студенты выполняют курсовые работы на объектно-ориентированных

языках программирования C#, Java (при желании и договоренности с руководителем, работа может быть выполнена на других объектно-ориентированных языках).

Руководят работой, консультируют, помогают уточнить план, список литературы, контролируют ход выполнения работы преподаватели кафедры информационных технологий и статистики.

## Организация выполнения курсовой работы

### Структура и требования к содержанию курсовой работы

Сущность темы должна быть изложена четко и последовательно, каждый последующий раздел должен быть логическим продолжением предыдущего, вытекать из него и быть с ним взаимосвязанным. Общий объем работы должен составлять 25 – 35 страниц, включая список использованных источников (не включая приложения). Курсовая работа должна быть оформлена в соответствии с Положением о требованиях к оформлению рефератов, отчетов по практике, контрольных, курсовых и дипломных работ, утвержденным Решением Совета по учебно-методическим вопросам и качеству образования УрГЭУ от 17.03.2011 г.

Курсовая работа включает:

титульный лист;

содержание;

введение;

основную часть, соответствующую теме курсовой работы;

заключение;

список использованных источников;

приложения.

Титульный лист должен содержать все необходимые идентификационные признаки исполнителя (см. приложение А).

В содержании последовательно перечисляются заголовки разделов курсовой работы, указываются номера страниц, с которых они начинаются.

Во введении к курсовой работе необходимо:

- обосновать актуальность темы курсовой работы;
- сформулировать цель курсовой работы и ее задачи;
- Привести краткую аннотацию работы.

Объем введения – 1-2 страницы текста.

В основной части работы раскрывается содержание заявленной темы.

В заключении приводятся основные выводы по курсовой работе, подтверждающие выполнение поставленных задач и отражающие полученные в курсовой работе результаты.

Объем заключения – 1-2 страницы.

Список использованных источников должен содержать перечень использованных при написании курсовой работы литературных источников с их полным описанием по требованиям стандартов. В список в обязательном порядке включаются источники, материалы которых использовались в процессе рассмотрения разделов, указанных в содержании работы (ссылки на Интернет – источники обязательны). Список использованных источников должен включать не менее 20 наименований, в том числе не менее 2 наименований на иностранном языке. В тексте курсовой работы обязательно должны находиться ссылки на указанную в данном списке литературу (в квадратных скобках).

В приложения выносятся расчеты, систематизирующие первичные материалы, а также иллюстрационный материал, не носящий принципиального характера, но требуемый логикой изложения, отчетные формы документов, листинги программ с комментариями и другие источники информации.

### *Содержание*

Слово «содержание» записывают в виде заголовка симметрично тексту прописными буквами. В содержании работы указывается перечень всех глав и параграфов курсовой работы, а также номера страниц, с которых начинается каждый из них. Главы в курсовой работе должны иметь в пределах всей работы порядковые номера, обозначенные арабскими цифрами. Параграфы каждой главы должны иметь нумерацию в пределах каждой главы. Номер параграфа состоит из номера главы и

непосредственно номера параграфа в данной главе, отделенного от номера главы точкой.

### *Введение*

Во введении должно присутствовать: актуальность темы, цель разработки системы, назначение, решаемые задачи, краткое описание содержания курсовой работы. Введение не должно превышать двух страниц.

### *Основная часть работы*

Основная часть оформляется в виде следующих глав и параграфов.

Первая глава является аналитической и посвящена анализу предметной области. Первый параграф посвящен *описанию предметной области*: чему посвящена предметная область, какие в ней есть термины и понятия, субъекты и объекты, способы взаимодействия субъектов, способы использования объектов, закономерности. Отдельный параграф должен быть посвящен выделению *словаря предметной области*: что входит в словарь предметной области, отдельно выделив список существительных и список глаголов, которые могут быть связаны с существительными. Третий параграф посвящен *анализу существующих программных решений* и известных подходов к решению подобных задач. Четвертый параграф посвящен подготовке технического задания на разработку программного обеспечения. *Техническое задание* в обязательном порядке включает в себя функциональные требования (с диаграммой вариантов использования и описанием вариантов использования), требования к экранным формам, модель данных и нефункциональные требования. Другие пункты технического задания могут включаться дополнительно.

Вторая глава курсовой работы является проектной и посвящена проектированию и разработке программного обеспечения. Первый параграф должен содержать описание процесса *проектирования и разработки классов*. Входной информацией являются определенные на этапе анализа задач объекты, общая характеристика программы. Проектирование системы классов начинается с обработки словаря предметной области. Эта обработка состоит в выявлении того, какие слова соответствуют объектам, классам, свойствам и операциям. Список



существительных служит основой для выделения классов и их свойств, а список глаголов – для определения операций. Следующий шаг состоит в том, чтобы определить, какой из классов какие свойства и функции содержит. Следует обратить внимание на то, что эти наборы у разных классов могут «пересекаться». Ещё одним вопросом, требующим решения на данном шаге, является выявление отношений между классами. Речь идёт об отношениях наследования и включения. Следует обратить внимание, что понятие «наследование» чаще всего возникает тогда, когда разные классы обладают частично схожими наборами свойств и методов. При составлении словаря предметной области далеко не всегда в список могут попасть понятия, которым можно сопоставить базовые классы в иерархии наследования. Поэтому следует внимательно проанализировать список классов, свойств, методов, их соответствие, и, возможно, выделить ряд новых классов, связанных с имеющимися отношениями наследования и включения. На данном этапе можно уже учитывать не только законы предметной области, но и такие принципы объектно-ориентированного подхода как абстракция, инкапсуляция, полиморфизм. Завершается этап тем, что разрабатываются полные спецификации базовых классов системы. При этом характеристики объектов вписываются в поля классов, а на основании операций разрабатываются методы и определяются способы доступа (public, private, protected). Следует иметь в виду, что все имена классов, полей и методов должны быть осмысленными, начинаться с заглавной буквы. На этапе проектирования для именованя классов, полей и методов рекомендуется использовать русские слова, однако, на этапе разработки предпочтительным является английский язык. Большим плюсом к проекту будет являться использование известных паттернов (шаблонов) проектирования, таких как: одиночка (singleton), прототип (prototype), строитель (builder), абстрактная фабрика (abstract factory), адаптер (adapter), декоратор (decorator), заместитель (proxy), мост (bridge), компоновщик (composite), фасад (facade), итератор (iterator), команда (command), наблюдатель (observer), стратегия (strategy), посетитель (visitor), посредник (mediator), состояние (state). Завершается этот этап подготовкой и представлением в этом разделе UML- диаграмм классов.

Второй параграф посвящен *проектированию пользовательского интерфейса*. На этом этапе разрабатываются структурная схема пользовательского интерфейса программы, детали управления системой (пользовательские истории) и дизайн-макеты основных окон графического интерфейса. Структурная схема позволяет выверить все детали проекта, определить взаимоотношения между отдельными частями программы, а также определяет содержание программных сообщений. На основании этой схемы в дальнейшем можно построить схему движения информационных потоков, диаграмму взаимодействия классов и т.д.

Третий параграф посвящен *описанию разработанных алгоритмов и модулей*. В данном параграфе приводится иерархическая структура приложения и схема связности модулей, а также описание наиболее важных (сложных) алгоритмов, разработанных для данного программного комплекса. Алгоритмы можно описать с использованием классического языка блок-схем, языка программирования (с комментариями) или с использованием нотации UML-диаграмм (activity, sequence). В курсовой работе в обязательном порядке должны быть использованы классические алгоритмы обработки структур данных (сортировка, поиск, списки, деревья, графы и пр.) и собственные разработанные алгоритмы. Исходный код разработанных программ следует включать в приложения к курсовой работе. Четвертый параграф содержит описание процесса тестирования программного комплекса, включая описание тестов, их анализ, достоинства и недостатки разработки. Пятый параграф содержит краткое *руководство пользователя* по эксплуатации разработанной системы.

Названия глав и параграфов должны быть, по возможности, отличными от предложенных и отражать тематику курсовой работы. В некоторых работах могут присутствовать не все вышеперечисленные параграфы, тогда как в некоторых работах могут добавляться дополнительные параграфы, если того требует тематика курсовой работы.

*Заключение*

В заключении логически последовательно излагаются основные результаты, использованные методы, оценка полноты решений поставленных задач, значимость работы, область применения полученных результатов.

#### *Список использованных источников*

Список использованных источников является составной частью работы и отражает степень изученности рассматриваемой проблемы. При этом в список включаются, как правило, не только те источники, на которые в работе имеются библиографические ссылки, но и те, которые были изучены при исследовании темы работы: список использованной литературы, фундаментальные труды, книги, публикации в периодической печати, Интернет – источники. При этом, в основной части работы должны быть представлены ссылки на указанные в данном списке источники.

#### *Приложения*

В приложении в обязательном порядке должен быть представлен полный листинг разработанной программы с комментариями.

Также, в приложения следует относить вспомогательный материал, который при включении в основную часть работы загромождает текст. К вспомогательному материалу относятся промежуточные расчеты, таблицы вспомогательных цифровых данных, программный код, инструкции, методики, распечатки текстов программ для компьютеров, иллюстрации вспомогательного характера, заполненные формы отчетности и других документов.

### **Порядок представления на проверку и защита курсовой работы**

Защита курсовой работы проводится в следующем порядке:

а) не позднее, чем за две недели до защиты курсовой работы студентом руководитель рассматривает код написанной программы (или демонстрационные примеры использования выбранной технологии) и допускает до защиты;

б) студент оформляет курсовую работу и готовит доклад для выступления;

в) защита выполняется публично, путем доклада перед комиссией, в которой обязательно присутствует преподаватель дисциплины и другие преподаватели. В результате защиты оценивается сложность работы, качество оформления материалов,

выполненный доклад. На основании усреднения баллов членов комиссии выставляется оценка, даются рекомендации по публикации результатов, размещении результатов проекта на сайте кафедры.

После защиты курсовой работы студент получает окончательную оценку, которая проставляется в зачетной книжке и в ведомости.

Курсовая работа позволяет студенту обобщить и систематизировать полученные ранее знания, углубить их в области экономических дисциплин, развить способности к творческому мышлению в сфере управления предприятием, а также является фундаментом для подготовки выпускной квалификационной работы.

### Примерные темы курсовых работ

- 1 Разработка клиент-серверного игрового приложения
- 2 Разработка музыкального плеера
- 3 Разработка игры в жанре «Платформер»
- 4 Разработка прикладной программы для контент-менеджеров
- 5 Разработка приложения для автоматизированного создания сайтов
- 6 Разработка настольной игры
- 7 Разработка приложения для обмена сообщениями с шифрованием
- 8 Разработка сервисного приложения с использованием API-интерфейсов
- 9 Разработка программы для автоматизации учета товаров на складе
- 10 Разработка программы-симулятора
- 11 Разработка автоматизированной системы управления персоналом
- 12 Разработка программы для решения систем уравнений
- 13 Разработка программы звукозаписи
- 14 Разработка системы управления складом предприятия
- 15 Разработка прикладного решения для поиска и удаления дубликатов файлов
- 16 Разработка приложения для поддержания здорового образа жизни
- 17 Разработка программы для конвертации валют
- 18 Разработка графического редактора

19 Разработка многопользовательской онлайн игры

20 Разработка программного комплекса учета приема-выдачи оборудования с использованием штрихового кодирования

## Типовая структура курсовой работы

### 1. Аналитическая часть

1.1. Описание предметной области

1.2. Словарь предметной области

1.3. Анализ существующих программных решений

1.4. Техническое задание

1.4.1. Функциональные требования

1.4.2. Требования к экранным формам

1.4.3. Модель данных

1.4.4. Нефункциональные требования

### 2. Проектная часть

2.1. Проектирование и разработка классов

2.2. Проектирование и разработка базы данных *(если имеется)*

2.3. Проектирование пользовательского интерфейса

2.4. Описание разработанных алгоритмов и программных модулей

2.4.1. Иерархическая структура приложения

2.4.2. Алгоритмы и математическое обеспечение

2.4.3. Описание основных программных модулей

2.5. Тестирование программного комплекса

2.5.1. Описание методики тестирования

2.5.2. Результаты тестов и их анализ

2.6. Руководство пользователя

## Критерии оценки курсовых работ

Критерий оценки	Оценка «отлично»	Оценка «хорошо»	Оценка «удовлетворительно»
1. Актуальность выбранной темы, обоснованность значимости проблемы для объекта исследования	В работе четко обоснована актуальность темы КР с точки зрения значимости проблемы исследования. Формулировки точные и аргументированные.	В работе недостаточно полно обоснована актуальность темы КР и значимость проблемы исследования. Допускаются отдельные недочеты в формулировках.	Актуальность темы КР обоснована не точно. Значимость проблемы исследования сформулирована фрагментарно.
2. Теоретическая и практическая значимость исследования;	В работе обоснована теоретическая значимость исследования, отражена его связь с задачами профессиональной деятельности. Работа содержит рекомендации по использованию результатов проведенного исследования на практике	В работе не четко обоснована теоретическая значимость исследования, отражена связь исследования с задачами профессиональной деятельности.  Работа содержит элементы рекомендаций по использованию результатов проведенного исследования на практике	В работе не обоснована теоретическая значимость исследования, связь исследования с задачами профессиональной деятельности не четко определена.
3. Четкость формулировок цели и задач исследования, методическая грамотность	Цель и задачи исследования грамотно сформулированы, структура работы им полностью соответствует. Правильно выбраны и применены необходимые методы исследования.	Цель и задачи грамотно сформулированы, структура работы в основном им соответствуют. Имеются незначительные ошибки в выборе и/или применении методов исследования.	Цель и задачи сформулированы недостаточно четко, слабо связаны со структурой работы. Имеются грубые ошибки в выборе и/или применении методов исследования.
4. Логичность изложения материала. Наличие аргументированных выводов по результатам работы, их соответствие целевым установкам	В работе имеется четкая структура, внутреннее единство и композиционная целостность, логическая последовательность изложения материала. Сделаны аргументированные выводы по результатам работы, они соответствуют целевым установкам.	Материал изложен логично и последовательно, но имеются недочеты в структуре работы. Сделаны выводы по результатам работы, но они не всегда аргументированы. Выводы в основном соответствуют целевым установкам.	В работе отсутствует внутреннее единство, имеются нарушения в логике и последовательности изложения материала. Выводы поверхностные, не всегда соответствуют целевым установкам
5. Наличие обоснованных предложений по совершенствованию предметной области, изложение своего	В работе содержатся самостоятельно сформулированные предложения по совершенствованию	В работе сформулированы предложения по совершенствованию деятельности организации, однако обоснования	Рекомендации по совершенствованию деятельности организации носят формальный характер. Видение

Критерий оценки	Оценка «отлично»	Оценка «хорошо»	Оценка «удовлетворительно»
видения перспектив дальнейшего исследования проблемы	деятельности в организации, имеются обоснования, намечены пути дальнейшего исследования темы. (Возможно наличия акта внедрения)	выполнены недостаточно корректно. Видение перспектив дальнейших исследований не структурированное.	перспектив дальнейших исследований отсутствует.
6. Уровень языковой и стилистической грамотности.	Высокий уровень языковой и стилистической грамотности. В работе отсутствуют речевые и орфографические ошибки. Автор свободно владеет деловым стилем речи	В работе допущены некоторые стилистические и речевые погрешности, при этом автор хорошо владеет деловым стилем речи	Недостаточное владение деловым стилем речи В работе имеются различного рода ошибки, опечатки исправлены не полностью.
7. Качество оформления работы	Работа оформлена в соответствии со всеми требованиями, предъявляемыми к КР.	Имеются не значительные недочеты в оформлении	Много недочетов в оформлении.
8. Качество презентационных материалов и устного выступления	Лаконичный и содержательный доклад отражающий основные положения и результаты исследования. Соблюдение установленного регламента. Ясные и четкие ответы на задаваемые вопросы и высказываемые замечания. Свободная ориентация в теме	Недостаточное освещение проблем исследования, некоторые сложности в формулировке главных выводов. Нарушение временного регламента незначительное. Ясные и четкие ответы на задаваемые вопросы и высказываемые замечания. Свободная ориентация в теме.	Доклад не дает представления о содержании и результатах исследования. Несоблюдение временного регламента. Затруднения в ответах на вопросы, неточные формулировки.

## Список использованных источников

### Основная литература

1. Плещев, В. В. Основы программирования на языках С++ и С# с примерами и упражнениями [Текст] : учебное пособие / В. В. Плещев, Е. И. Шишков ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2018. - 286 с. <http://lib.usue.ru/resource/limit/ump/18/p490708.pdf> 40экз.
2. Кислицын, Е. В. Разработка приложений на языке Java [Текст] : учебное пособие / Е. В. Кислицын, Е. И. Шишков ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2017. - 86 с. <http://lib.usue.ru/resource/limit/ump/17/p488938.pdf> 30экз.

3. Колдаев, В. Д. Структуры и алгоритмы обработки данных [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальностям 230105 "Программное обеспечение вычислительной техники и автоматизированных систем", 230101 "Вычислительные машины, комплексы, системы и сети", 080801 "Прикладная информатика в экономике" / В. Д. Колдаев. - Москва : РИОР: ИНФРА-М, 2014. - 296 с. <http://znanium.com/go.php?id=418290>
4. Белов, В. В. Алгоритмы и структуры данных [Электронный ресурс] : учебник для студентов вузов, обучающихся по направлению подготовки 2.09.03.04 "Программная инженерия" (квалификация - Бакалавр) / В. В. Белов, В. И. Чистякова. - Москва : КУРС: ИНФРА-М, 2019. - 240 с. <http://znanium.com/go.php?id=978314>

#### **Дополнительная литература**

1. Хорев, П. Б. Объектно-ориентированное программирование с примерами на С# [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по направлениям 01.03.02 "Прикладная математика и информатика" и 09.00.00 "Информатика и выч. техника" / П. Б. Хорев. - Москва : ФОРУМ: ИНФРА-М, 2016. - 200 с. <http://znanium.com/go.php?id=529350>
2. Жеребцов, А. С. Объектно-ориентированный анализ и программирование [Текст] : учебное пособие / А. С. Жеребцов, С. Ф. Молодецкая ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2014. - 126 с. <http://lib.usue.ru/resource/limit/ump/14/p482025.pdf> 10экз.
3. Шилдт, Г. Java 8 [Текст] : руководство для начинающих / Герберт Шилдт ; [пер. с англ. и ред. А. Г. Гузикевича]. - 6-е изд. - Москва : Вильямс, 2017. - 712 с. 30экз.
4. Дроздов, С. Н. Структуры и алгоритмы обработки данных [Электронный ресурс] : Учебное пособие / С. Н. Дроздов ; М-во образования и науки Рос. Федерации, Юж. федер. ун-т, Инжен.-технол. акад. - Таганрог : Издательство ЮФУ, 2016. - 228 с. <http://znanium.com/go.php?id=991928>
5. Вирт, Н. Алгоритмы и структуры данных. Новая версия для Оберона [Текст] : [учебник] / Никлаус Вирт ; пер. с англ. под ред. Ткачева Ф. В. - Москва : ДМК [Пресс], 2014. - 272 с. 5экз.
6. Седжвик, Р. Алгоритмы на Java [Текст] : научное издание / Роберт Седжвик, Кевин Уэйн ; [пер. с англ. А. А. Моргунова; под ред. Ю. Н. Артеменко]. - 4-е изд. - Москва : Вильямс, 2017. - 843 с. 20экз.

## **Основные требования по оформлению курсовой работы**

### *Общие требования*

Курсовая работа должна быть выполнена на одной стороне листа белой бумаги формата А4. Текст работы должен быть подготовлен компьютерным способом в текстовом редакторе MS Word (тип шрифта –Times new roman; размер шрифта – 14 пт; межстрочный интервал – 1,5; цвет шрифта – чёрный, равномерная плотность, контрастность и четкость изображения) и распечатан.

Разрешается использовать компьютерные возможности акцентирования внимания на определенных терминах и формулах, применяя курсив, полужирный шрифт не применяется. Не допускается цветовое оформление заголовков рисунков, предложений и отдельных слов, а также вложение листов в отдельные файлы.

В печатном тексте допускается вписывание отдельных слов, формул, знаков только черными чернилами или черной тушью.



Текст работы следует печатать, соблюдая поля (правое – 10 мм, верхнее и нижнее – 20 мм, левое – 30 мм), абзацный отступ – 1,25 см. Рамки, ограничивающие текст, не вычерчиваются.

Описание работы следует излагать научным языком, просто, сжато и логически последовательно.

Следует избегать повторов, лишних слов, затрудняющих восприятие текста. В тексте всей работы необходимо соблюдать единообразие терминов, обозначений, символов.

Сокращение русских слов и словосочетаний допускается при условии соблюдения требований ГОСТ 7.12–93 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила».

В тексте курсовой работы допускаются общепринятые сокращения и аббревиатуры, установленные правилами орфографии и соответствующими нормативными документами, например: год – г., годы – гг., и так далее – и т. д., метр – м, тысяч – тыс., миллион – млн, миллиард – млрд, триллион – трлн, страница – с., Российская Федерация – РФ, общество с ограниченной ответственностью – ООО.

При использовании авторской аббревиатуры необходимо при первом ее упоминании дать полную расшифровку, например: «... Уральский государственный экономический университет (далее – УрГЭУ)...».

Не допускается использование сокращений и аббревиатур в заголовках письменной работы, глав и параграфов.

В тексте следует избегать личных местоимений, заменяя их безличными формами (вместо «я считаю» следует писать «автор считает» или «считается»).

При необходимости в тексте курсовой работы могут быть приведены перечисления. Перед каждым элементом перечисления следует ставить дефис (иные маркеры не допустимы!).

Например:

«...закключение содержит:

- краткие выводы;
- оценку решений;
- разработку рекомендаций».

При необходимости ссылки в тексте работы на один из элементов перечисления вместо дефиса ставятся строчные буквы в порядке русского алфавита, начиная с буквы а (за исключением букв ё, з, й, о, ч, ь, ы, ь). Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, а запись производится с абзацного отступа. Например:

а) ...;

б) ...;

1) ...;

2) ...;

в) ... .

Страницы курсовой работы нумеруются арабскими цифрами внизу, по центру. Номер страницы не ставится на титульном листе и содержании, но они входят в общую нумерацию страниц так же, как и приложения. Иллюстрации по тексту работы, расположенные на отдельных листах, и страницы приложений нумеруются.

## Оформление структурных элементов курсовой работы

Структурными элементами курсовой работы являются: содержание, введение, основная часть, заключение, список использованных источников.

Каждый структурный элемент курсовой работы необходимо начинать с новой страницы. Следующий параграф внутри одной главы начинается через 2 межстрочных интервала на том же листе, где закончился предыдущий.

Расстояние между заголовком структурного элемента и текстом, заголовками главы и параграфа, заголовком параграфа и текстом составляет 2 межстрочных интервала.

Наименования структурных элементов курсовой работы («СОДЕРЖАНИЕ», «ВВЕДЕНИЕ», «ЗАКЛЮЧЕНИЕ» и т.д.) служат заголовками структурных элементов. Данные наименования пишутся по центру страницы прописными (заглавными) буквами без точки в конце и без подчеркивания.

Главы и параграфы основной части должны иметь заголовки. Их следует нумеровать арабскими цифрами и записывать по центру страницы прописными

(заглавными) буквами без точки в конце, не подчеркивая. Номер главы указывается цифрой (например, 1, 2, 3), номер параграфа включает номер главы и порядковый номер параграфа, разделенные точкой (например, 1.1, 2.1, 3.3). После номера главы и параграфа в тексте точку не ставят. Если заголовок состоит из двух предложений, их разделяют точкой. Переносы слов в заголовках не допускаются. Не допускается писать заголовок параграфа на одном листе, а его текст – на другом.

## Использование и оформление рисунков

В курсовой работе для наглядности, уменьшения физического объема сплошного текста следует использовать иллюстрации – графики, схемы, диаграммы, чертежи, рисунки и фотографии. Все иллюстрации именуются рисунками. Их количество зависит от содержания работы и должно быть достаточно для того, чтобы придать ей ясность и конкретность.

Рисунки должны быть созданы с помощью инструментов Microsoft Word, возможно использование цвета. При цветном исполнении рисунков следует использовать принтер с возможностью цветной печати. При использовании в рисунках черно-белой печати следует применять черно-белую штриховку элементов рисунка.

На все рисунки должны быть даны ссылки в тексте работы, например: «... в соответствии с рисунком 2 ...» или «... тенденцию к снижению (рисунок 2)».

Рисунки следует располагать в работе непосредственно после текста, в котором они упоминаются впервые (при наличии достаточного пространства для помещения рисунка со всеми поясняющими данными), или на следующей странице. Если рисунок достаточно большой по размерам или если в нем имеется много детализированной информации, его можно размещать на отдельном листе. Допускается поворот рисунка по часовой стрелке (если он выполнен на отдельном листе). Рисунки, размеры которых больше формата А4, учитывают как одну страницу и размещают в приложениях.

Рисунки, за исключением рисунков в приложениях, следует нумеровать арабскими цифрами сквозной нумерацией по всей работе. Каждый рисунок (схема, график, диаграмма), обозначаемый словом «Рисунок», должен иметь заголовок и

подписываться следующим образом – посередине строки без абзацного отступа, например:

### Рисунок 1 – Структура администрации района

Если на рисунке отражены показатели, то после заголовка рисунка через запятую указывается единица измерения, например:

### Рисунок 1 – Структура издержек, %

Рисунки каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения (например, рисунок А.3).

Если рисунок взят из первичного источника без авторской переработки, следует сделать ссылку, например:

### Рисунок 2 – Система работы с кадрами [8, с. 15]

Если рисунок является авторской разработкой, необходимо после заголовка рисунка поставить знак сноски и указать в форме подстрочной сноски внизу страницы, на основании каких источников он составлен, например:

---

<sup>1</sup> Составлено автором по: [15, 23, 42].

При необходимости между рисунком и его заголовком помещаются поясняющие данные (подрисуночный текст), например, легенда.

## Использование и оформление таблиц

В курсовой работе фактический материал в обобщенном и систематизированном виде может быть представлен в виде таблиц для наглядности и удобства сравнения показателей.

На все таблицы должны быть ссылки в работе. При ссылке следует писать слово «таблица» с указанием ее номера, например: «...в таблице 2 представлены ...» или «... характеризуется показателями (таблица 2)».

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

Таблицы, за исключением таблиц в приложениях, следует нумеровать арабскими цифрами сквозной нумерацией по всей работе. Каждая таблица должна иметь заголовок, который должен отражать ее содержание, быть точным, кратким. Заголовок таблицы следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тире, например:

Таблица 1 – Динамика показателей за 2010–2011 гг.

Если таблица взята из первичного источника без авторской переработки, следует сделать ссылку, например:

Таблица 1 – Источники набора персонала [15, с. 35]

Если таблица является авторской разработкой, необходимо после заголовка таблицы поставить знак сноски и указать в форме подстрочной сноски внизу страницы, на основании каких источников она составлена, например:

---

<sup>1</sup> Составлено автором по: [1, 3, 10].

Располагают таблицы на странице обычно вертикально. Помещенные на отдельной странице таблицы могут быть расположены горизонтально, причем графа с наименованиями показателей должна размещаться в левой части страницы. Таблицы ограничивают линиями.

Таблицу с большим числом строк допускается переносить на другую страницу. При переносе части таблицы на другую страницу слово «Таблица» указывают один раз слева над первой частью таблицы. На странице, на которую перенесена часть таблицы, слева пишут «Продолжение таблицы» или «Окончание таблицы» с указанием номера таблицы и повторением шапки таблицы.

Если таблица переносится, то на странице, где помещена первая часть таблицы, нижняя ограничительная линия таблицы не проводится. Это же относится к странице (страницам), где помещено продолжение (продолжения) таблицы. Нижняя

ограничительная линия таблицы проводится только на странице, где помещено окончание таблицы.

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

Примечания к таблице (подтабличные примечания) размещают непосредственно под таблицей в виде: а) общего примечания; б) сноски; в) отдельной графы или табличной строки с заголовком. Выделять примечание в отдельную графу или строку целесообразно лишь тогда, когда примечание относится к большинству строк или граф. Примечания к отдельным заголовкам граф или строк следует связывать с ними знаком сноски. Общее примечание ко всей таблице не связывают с ней знаком сноски, а помещают после заголовка «Примечание» или «Примечания», оформляют как внутритекстовое примечание.

Допускается применять размер шрифта в таблице меньший, чем в тексте работы, но не менее 10 пт.

Если все показатели, приведенные в графах таблицы, выражены в одной и той же единице измерения, то ее обозначение необходимо помещать над таблицей справа. Если показатели таблицы выражены в разных единицах измерения, то обозначение единицы измерения указывается после наименования показателя через запятую. Допускается при необходимости выносить в отдельную графу обозначения единиц измерения.

Текст, повторяющийся в строках одной и той же графы и состоящий из одиночных слов, чередующихся с цифрами, заменяют кавычками. Если повторяющийся текст состоит из двух или более слов, то при первом повторении его заменяют словами «То же», а далее – кавычками. Если предыдущая фраза является частью последующей, то допускается заменить ее словами «То же» и добавить дополнительные сведения. При наличии горизонтальных линий текст необходимо повторять. Если в ячейке таблицы приведен текст из нескольких предложений, то в последнем предложении точка не ставится.

Заменять кавычками повторяющиеся в таблице цифры, математические знаки, знаки процента и номера, обозначения нормативных материалов, марок материалов не допускается.

При отсутствии отдельных данных в таблице следует ставить прочерк (тире). Цифры в графах таблиц должны проставляться так, чтобы разряды чисел во всей графе были расположены один под другим, если они относятся к одному показателю. В одной графе должно быть соблюдено, как правило, одинаковое количество десятичных знаков для всех значений величин.

Если таблицы размещены в приложении, их нумерация имеет определенные особенности. Таблицы каждого приложения нумеруют отдельной нумерацией арабскими цифрами. При этом перед цифрой, обозначающей номер таблицы в приложении, ставится буква соответствующего приложения, например:

Таблица В.1.– Динамика показателей за 2010–2011 гг.

Если в документе одна таблица, то она должна быть обозначена «Таблица 1» или «Таблица В.1», если она приведена в приложении (допустим, В).

## Использование и оформление формул

Для составления математических формул используется редактор формул Microsoft Word.

Формулы и уравнения следует выделять из текста в отдельную строку. Выше и ниже каждой формулы и уравнения необходимо оставлять не менее одной свободной строки.

Формулы следует нумеровать порядковой нумерацией в пределах всей работы арабскими цифрами в круглых скобках в крайнем правом положении на строке, например:

$$R = X_{\max} - X_{\min},$$

(4)

Формулы, размещаемые в приложениях, должны нумероваться отдельной нумерацией арабскими цифрами в пределах каждого приложения с добавлением перед каждой цифрой обозначения приложения, например, формула (В.1).

Ссылки в тексте на порядковые номера формул дают в скобках. Например: «.. в формуле (1)...».





}

}

## Оформление примечаний и ссылок

При необходимости пояснить содержание текста, таблицы или иллюстрации в курсовой работе следует помещать примечания. Их размещают непосредственно в конце страницы, таблицы, иллюстрации, к которым они относятся, и печатают с прописной буквы с абзацного отступа после слова «Примечание» или «Примечания». Если примечание одно, то после слова «Примечание» ставится тире и примечание печатается с прописной буквы. Одно примечание не нумеруют. Если их несколько, то после слова «Примечания» ставят двоеточие и каждое примечание печатают с прописной буквы с новой строки с абзацного отступа, нумеруя их по порядку арабскими цифрами.

Цитаты, а также все заимствованные из печати данные (нормативы, цифры и др.), должны иметь библиографическую ссылку на первичный источник. Ссылка ставится непосредственно после того слова, числа, символа, предложения, по которому дается пояснение, в квадратных скобках. В квадратных скобках указывается порядковый номер источника в соответствии со списком использованных источников и номер страницы, с которой взята информация, например: [3, с. 15].

Приводимые в работе цитаты должны быть по возможности краткими. Если цитата полностью воспроизводит предложение цитируемого текста, она начинается с прописной буквы. Если цитата включена на правах части в предложение авторского текста, она пишется со строчной буквы. Если в цитату вошла только часть предложения цитируемого источника, то либо после кавычки ставится многоточие и цитата начинается с маленькой буквы, либо цитата начинается с большой буквы и заканчивается многоточием, например: Ф.Котлер подчеркивал, что современный маркетинг «...все в большей степени ориентируется на удовлетворение потребностей индивидуального потребителя» [26, с. 84].

## Правила оформления списка использованных источников

Список должен содержать сведения об источниках, использованных при написании курсовой работы. Сведения об источниках приводятся в следующем порядке:

- официальные материалы;
- книги, статьи, материалы конференций и семинаров;
- статистические сборники, инструктивные материалы, методические рекомендации, реферативная информация, нормативно-справочные материалы;
- иностранная литература;
- интернет-сайты.

В списке использованных источников применяется сквозная нумерация с применением арабского алфавита. Все объекты печатаются единым списком, группы объектов не выделяются.

Объекты описания списка должны быть обозначены терминами в квадратных скобках<sup>1</sup>:

- [Видеозапись];
- [Мультимедиа];
- [Текст];
- [Электронный ресурс].

При занесении источников в список литературы следует придерживаться установленных правил их библиографического описания. Например:

*Официальные материалы.* В начале списка дается перечень использованных нормативных правовых актов федерального уровня в следующем порядке: международные нормативно-правовые акты, Конституция, кодексы, федеральные законы, указы Президента РФ, постановления Правительства РФ, нормативно-правовые акты иных федеральных органов государственной власти. Нормативные правовые акты одного уровня располагаются в хронологическом порядке, от принятых в более ранние периоды к принятым в более поздние периоды.

После федеральных нормативно-правовых актов перечисляются нормативно-правовые акты регионального, а затем муниципального уровней в том же порядке.

---

<sup>1</sup> Полный перечень см. в: Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления [Текст]: ГОСТ 7.1-2003.

Примеры оформления нормативно-правовых актов:

1. Об общих принципах организации законодательных (представительных) и исполнительных органов власти субъектов Российской Федерации [Текст]: Федеральный закон РФ от 06.10.1999. N 184-ФЗ //Собрание законодательства РФ. –1999.– №43.

2. О порядке разработки и утверждения административных регламентов исполнения государственных функций (предоставления государственных услуг) [Электронный ресурс]: Постановление Правительства РФ от 11.11.2005 г. N 679. – Доступ из справочно-правовой системы «КонсультантПлюс». – Режим доступа: <http://www.consultant.ru>.

*Книги, статьи, материалы конференций и семинаров.* Располагаются по алфавиту фамилии автора или названию, если книга печатается под редакцией. Например:

3. Боголюбов, А.Н. О вещественных резонансах в волноводе с неоднородным заполнением [Текст] / А.Н. Боголюбов, А.Л. Делицын, М.Д. Малых // Вестн. Моск. ун-та. Сер. 3, Физика. Астрономия. – 2001. – N 5. – С. 23–25.

4. Голубков, Е.П. Маркетинг как концепция рыночного управления [Текст] // Маркетинг в России и за рубежом. – 2001. – N 1. – С. 89–104.

5. Государственные и муниципальные финансы [Текст] : учебник / Под ред. проф. С.И. Лушина, проф. В.А. Слепова. – М.: Экономист, 2006. – 280 с.

6. Двинянинова, Г.С. Комплимент: Коммуникативный статус или стратегия в дискурсе [Текст] / Г.С. Двинянинова // Социальная власть языка: сб. науч. тр. / Воронеж. межрегион. ин-т обществ. наук, Воронеж. гос. ун-т, Фак. романо-герман. истории. – Воронеж, 2001. – С. 101–106.

7. История России [Текст] : учеб. пособие для студентов всех специальностей / В.Н. Быков [и др.] ; отв. ред. В.Н. Сухов ; М-во образования Рос. Федерации, С.-Петербург. гос. лесотехн. акад. – 2-е изд., перераб. и доп. / при участии Т.А. Суховой. – СПб.: СПбЛТА, 2001. – 231 с.

8. Семенов, В.В. Философия: итог тысячелетий. Философская психология [Текст] / В.В. Семенов; Рос. акад. наук, Пушин. науч. центр, Ин-т биофизики клетки, Акад. проблем сохранения жизни. – Пушино: ПНЦ РАН, 2000. – 64 с.

9. Черткова, Е.Л. Утопия как способ постижения социальной действительности [Электронный ресурс] / Е.Л. Черткова // Социемы: журнал Уральского гос. ун-та. – 2002. – N 8. – Режим доступа: <http://www2/usu.ru/philosoph/chertkova>.

10. Юридический советник [Электронный ресурс]. – 1 электрон. опт. диск (CD-ROM) : зв., цв. ; 12 см. – Прил.: Справочник пользователя [Текст] / сост. В.А. Быков. – 32 с.

*Статистические сборники, инструктивные материалы, методические рекомендации, реферативная информация, нормативно-справочные материалы.* Располагаются по алфавиту. Например:

11. Аппаратура радиоэлектронная бытовая. Входные и выходные параметры и типы соединений. Технические требования [Текст]: ГОСТ Р 517721-2001. – Введ. 2002-01-01. – М.: Изд-во стандартов, 2001. – 34 с.

12. Временные методические рекомендации по вопросам реструктуризации бюджетной сферы и повышения эффективности расходов региональных и местных бюджетов (Краткая концепция реструктуризации государственного и муниципального сектора и повышения эффективности бюджетных расходов на региональном и местном уровнях) [Текст]. – М.: ИЭПП, 2006. – 67 с.

13. Свердловская область в 1992-1996 годах [Текст]: Стат. сб. / Свердл. обл. комитет гос. статистики Госкомстата РФ. – Екатеринбург, 1997. – 115 с.

14. Социальное положение и уровень жизни населения России в 2010 г. [Текст]: Стат. сб. / Росстат. – М., 2002. – 320 с.

15. Социально-экономическое положение федеральных округов в 2010 г. [Электронный ресурс]. – Режим доступа: <http://www.gks.ru>.

*Иностранная литература.* Располагается по алфавиту. Например:

16. An Interview with Douglass C. North [Text] // The Newsletter of The Cliometric Society. – 1993. – Vol. 8. – N 3. – P. 23–28.

17. Burkhead, J. The Budget and Democratic Government [Text] / Lyden F.J., Miller E.G. (Eds.) / Planning, Programming, Budgeting. Markham : Chicago, 1972. 218 p.

18. Miller, D. Strategy Making and Structure: Analysis and Implications for Performance [Text] // Academy of Management Journal. – 1987. – Vol. 30. – N 1. – P. 45–51.

*Интернет-сайты.* Например:

19. Министерство финансов Российской Федерации: [Электронный ресурс]. – Режим доступа: <http://www.minfin.ru>

20. Российская книжная палата: [Электронный ресурс]. – Режим доступа: <http://www.bookchamber.ru>

### Правила оформления приложений

В приложения рекомендовано включать материалы, которые по каким-либо причинам не могут быть включены в основную часть: материалы, дополняющие работу; промежуточные математические доказательства, формулы и расчеты; таблицы вспомогательных цифровых данных; инструкции, методики, описания алгоритмов и программ задач, иллюстрации вспомогательного характера; нормативные акты, например, должностные инструкции. В приложения также включают иллюстрации, таблицы и распечатки, выполненные на листах формата А3.

Приложения оформляют как продолжение данного документа на последующих его листах после списка использованных источников.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ь (ПРИЛОЖЕНИЕ А, ПРИЛОЖЕНИЕ Б, ПРИЛОЖЕНИЕ В и т.д.). Допускается обозначение приложений буквами латинского алфавита, за исключением букв I и O. В случае полного использования букв русского и латинского алфавитов допускается обозначать приложения арабскими цифрами.

Само слово «ПРИЛОЖЕНИЕ» пишется прописными (заглавными) буквами.

Если в работе одно приложение, оно обозначается «ПРИЛОЖЕНИЕ А».

Каждое приложение следует начинать с новой страницы. При этом слово «ПРИЛОЖЕНИЕ» и его буквенное обозначение пишутся с абзацного отступа.

Приложение должно иметь заголовок, который записывают на следующей строке после слова «ПРИЛОЖЕНИЕ» с абзацного отступа. Заголовок пишется с прописной буквы.

В тексте работы на все приложения должны быть даны ссылки, например: «... в приложении Б...». Приложения располагают в порядке ссылок на них в тексте работы.

Текст каждого приложения, при необходимости, может быть разделен на разделы, подразделы, пункты, подпункты, которые нумеруют в пределах каждого приложения. Перед номером ставится обозначение этого приложения.

Приложения должны иметь общую с остальной частью документа сквозную нумерацию страниц.

## Оформление содержания

В содержании курсовой работы наименования структурных элементов указываются с левого края страницы, при этом первая буква наименования является прописной (заглавной), остальные буквы являются строчными, например:

Введение	3
1. Постановка задачи	5
1.1 Цели и задачи разрабатываемой системы	5
1.2 Входная, нормативно-справочная и выходная информация	7
1.3 Перечень используемых функций	9
2. Теоретические аспекты создания клиент-серверных приложений	10
2.1 Клиент-серверные приложения: основные положения	10
.....	
3. Алгоритмы разработки игровых приложений	15
.....	
Заключение	30
Список использованных источников	32
Приложение А	33

*Не ставятся точки между названием элемента содержания и номером страницы.*

