

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное автономное образовательное учреждение  
высшего образования**  
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

Методические указания к самостоятельной работе  
для студентов направления  
09.03.02 «Информационные системы и технологии»  
**по дисциплине**  
**«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»**

Невинномысск, 2021



## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1. ЦЕЛИ И ЗАДАЧИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	4
2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ .....	6
3. ТЕМАТИЧЕСКИЙ ПЛАН ДИСЦИПЛИНЫ .....	7
4. ВОПРОСЫ И ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ .....	9
5. ПОДГОТОВКА К ЗАНЯТИЯМ .....	21
5.1. Подготовка к лекциям .....	21
5.2. Подготовка к практическим занятиям .....	23
5.3. Самостоятельное изучение материала тем .....	25
5.4. Подготовка к экзамену .....	28
6. ПЛАН-ГРАФИК ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	30
7. КОНТРОЛЬНЫЕ ТОЧКИ И ОТЧЕТНОСТЬ ПО НИМ .....	32
ЛИТЕРАТУРА .....	34

## **ВВЕДЕНИЕ**

Целью освоения дисциплины «Объектно-ориентированное программирование» является формирование навыков выполнения работ по созданию (модификации) и сопровождению ИС, автоматизирующих задач организационного управления и бизнес-процессов.

Основными задачами дисциплины является: формирование представлений об общей методологии и средствах технологии объектно-ориентированного программирования; углубленная подготовка студентов в области применения технологии объектно-ориентированного программирования.

Методические указания предназначены для выполнения самостоятельной работы по дисциплине «Объектно-ориентированное программирование» с учетом требований ФГОС ВО для направления подготовки 09.03.02 «Информационные системы и технологии». Они способствуют лучшему усвоению студентами теоретических положений и обеспечивают приобретение практических навыков по исследованию профессионально-ориентированных информационных систем.

### **1. ЦЕЛИ И ЗАДАЧИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ**

Самостоятельная работа студентов (далее — СРС) является неотъемлемой составляющей образовательного процесса в Университете и является обязательной для каждого студента. Основная цель СРС — освоение в полном объеме образовательной программы и последовательное формирование компетенций эффективной самостоятельной профессиональной (практической и научно-теоретической) деятельности. Самостоятельная работа конкретна по своей предметной направленности и сопровождается непрерывным контролем и оценкой ее результатов.

Количество часов, отводимое на самостоятельную работу, определяется учебным планом направления подготовки 09.03.02.

Содержательно самостоятельная работа студентов определяется ФГОС ВО направления подготовки 09.03.02, программой и учебно-методическим комплексом дисциплины «Объектно-ориентированное программирование».

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей дисциплины «Объектно-ориентированное программирование», объема часов на ее изучение, вида заданий для СРС, индивидуальных возможностей студентов и условий учебной деятельности.

Формы самостоятельной работы студентов определяются содержанием дисциплины «Объектно-ориентированное программирование», степенью подготовленности студентов. Они могут быть тесно связаны с теоретическим курсом и иметь учебный или учебно-исследовательский характер. Форму самостоятельной работы студентов определяют кафедра ИСЭА при разработке программы дисциплины «Объектно-ориентированное программирование».

Самостоятельная работа может осуществляться индивидуально или группами студентов в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений студентов.

СРС, не предусмотренная образовательной программой, учебным планом и учебно-методическими материалами, раскрывающими и конкретизирующими их содержание, осуществляется студентами инициативно, с целью реализации собственных учебных и научных интересов.

В учебном процессе выделяют аудиторную и внеаудиторную самостоятельную работу.

Аудиторная самостоятельная работа по дисциплине «Объектно-ориентированное программирование» выполняется на учебных занятиях (лекциях, лабораторных занятиях и консультациях) под руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа студентов выполняется во внеаудиторное время по заданию и при методическом руководстве и контроле преподавателя, но без его непосредственного участия. СРС включает в себя:

- подготовку к аудиторным занятиям (лекциям, практическим, лабораторным) и выполнение соответствующих заданий;
- работу над отдельными темами учебных дисциплин (модулей) в соответствии с учебно-тематическими планами;
- итоговых контрольных испытаний.

## 2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Наименование компетенций

Код	Формулировка
ПК-2	Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задач организационного управления и бизнес-процессов

Знания, умения, навыки и (или) опыт деятельности, характеризующие этапы формирования компетенций

Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций	Формируемые компетенции
<b>Знать:</b> методы и средства сборки модулей и компонент программного обеспечения; интерфейсы взаимодействия с внешней средой; интерфейсы взаимодействия внутренних модулей системы; методы и средства разработки процедур для развертывания программного обеспечения; методы и средства миграции и преобразования данных; языки, утилиты и среды программирования, средства пакетного выполнения процедур.	<b>ПК-2</b>
<b>Уметь:</b> писать программный код процедур интеграции программных модулей; использовать выбранную среду программирования для разработки процедур интеграции программных модулей; применять методы и средства сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов.	<b>ПК-2</b>
<b>Владеть:</b> методами разработки и документирования программных интерфейсов; методами разработки процедур сборки модулей и компонент программного обеспечения; разработка процедур развертывания и обновления программного обеспечения; разработка процедур миграции и преобразования (конвертации) данных; оценка и согласова-	<b>ПК-2</b>

ние сроков выполнения поставленных задач.

### 3. ТЕМАТИЧЕСКИЙ ПЛАН ДИСЦИПЛИНЫ

Наименование и содержание лекций при очной форме обучения

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	Объем часов	Интерактивная форма проведения
<b>6 семестр</b>			
1	Теоретические основы объектно-ориентированного программирования 1. Эволюция технологии разработки программных продуктов	1,50	
2	Теоретические основы объектно-ориентированного программирования 1. Основные принципы объектно-ориентированного программирования	1,50	
3	Теоретические основы объектно-ориентированного программирования 1. Классы и объекты	1,50	
4	Теоретические основы объектно-ориентированного программирования 1. Объектная декомпозиция	1,50	
5	Теоретические основы объектно-ориентированного программирования 1. Разработка программ с использованием объектно-ориентированной технологии	1,50	
6	Техника объектно-ориентированного программирования 1. Организация класса. Модификаторы доступа. Статические и экземплярные ресурсы. 2. Поля и методы класса. 3. Конструкторы и деструкторы.	1,50	
7	Техника объектно-ориентированного программирования	1,50	

	1. Композиция. Контейнерные классы.		
8	Техника объектно-ориентированного программирования 1. Наследование: родители и потомки. 2. Простое и множественное наследование.	1,50	
9	Техника объектно-ориентированного программирования 1. Полиморфизм: раннее и позднее связывание.	1,50	
10	Техника объектно-ориентированного программирования 1. Полиморфизм на основе абстрактных классов.	1,50	
11	Техника объектно-ориентированного программирования 1. Полиморфизм на основе виртуальных методов.	1,50	
12	Техника объектно-ориентированного программирования 1. Статическое делегирование методов. 2. Динамическое делегирование методов.	1,50	
13	Техника объектно-ориентированного программирования 1. Параметризованные классы.	1,50	
14	Техника объектно-ориентированного программирования 1. Интерфейсы.	1,50	
15	Техника объектно-ориентированного программирования 1. Генерация исключений. Порядок обработки исключений.	1,50	
16	Техника объектно-ориентированного программирования 1. Обрабатывающая конструкция try ... catch. 2. Финализирующая конструкция try ... finally.	1,50	
	<b>Итого за 6 семестр</b>	<b>24,00</b>	
	<b>Итого</b>	<b>24,00</b>	

Наименование практических занятий при очной форме обучения

№ Темы дисциплины	Наименование тем практических занятий	Объем часов	Интерактивная форма проведения
<b>6 семестр</b>			
<b>Тема 2. Техника объектно-ориентированного программирования</b>			
1	Определение класса	1,50	
2	Классы и функции	1,50	
3	Наследование	1,50	Выполнение творческого задания в составе малой группы
4	Полиморфизм	1,50	Выполнение творческого задания в составе малой группы
5	Массивы объектов, указатели и ссылки	1,50	
6	Шаблоны	1,50	
7	Исключительные ситуации	1,50	
8	Потоки	1,50	
9	Классы ввода/вывода	1,50	
10	Статические и константные члены, локальные классы	1,50	
11	Создание главного окна приложения	1,50	Выполнение творческого задания в составе малой группы
12	Создание многооконного приложения	1,50	Выполнение творческого задания в составе малой группы
13	Создание пользовательских диалоговых окон	1,50	Выполнение творческого задания в составе малой группы
14	Создание меню приложения	1,50	Выполнение творческого задания в составе малой группы
15	Создание элементов управления	1,50	Выполнение творческого задания в составе малой группы

16	Подготовка ADO.NET к работе в приложении	1,50	Выполнение творческого задания в составе малой группы
	<b>Итого за 6 семестр</b>	<b>24,00</b>	<b>12,0</b>
	<b>Итого</b>	<b>24,00</b>	<b>12,0</b>

Наименование и содержание лекций при заочной форме обучения

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	Объем часов	Интерактивная форма проведения
<b>3 курс</b>			
1	Теоретические основы объектно-ориентированного программирования 1. Основные принципы объектно-ориентированного программирования 2. Классы и объекты	1,50	
2	Техника объектно-ориентированного программирования 1. Организация класса. Модификаторы доступа. Статические и экземплярные ресурсы. 2. Поля и методы класса. 3. Конструкторы и деструкторы.	1,50	
3	Техника объектно-ориентированного программирования 1. Композиция. Контейнерные классы. 2. Наследование: родители и потомки. 3. Полиморфизм: раннее и позднее связывание.	1,50	
	<b>Итого за 3 курс</b>	<b>4,50</b>	
	<b>Итого</b>	<b>4,50</b>	

Наименование практических занятий при заочной форме обучения

№ Темы дисциплины	Наименование тем практических занятий	Объем часов	Интерактивная форма проведения
<b>3 курс</b>			
<b>Тема 2. Техника объектно-ориентированного программирования</b>			

1	Определение класса	1,50	
2	Создание главного окна приложения	1,50	
3	Создание элементов управления	1,50	Выполнение творческого задания в составе малой группы
	<b>Итого за 3 курс</b>	<b>4,50</b>	<b>1,50</b>
	<b>Итого</b>	<b>4,50</b>	<b>1,50</b>

#### 4. ВОПРОСЫ И ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ

### Вопросы для собеседования

#### Пороговый уровень

Тема 1 Теоретические основы объектно-ориентированного программирования

1. Программы с глобальными и локальными данными.
2. Структурное программирование: основные принципы, пошаговая детализация, процедурная декомпозиция; достоинства и недостатки.
3. Модульное программирование: интерфейс и реализация; достоинства и недостатки.
4. Объектно-ориентированное программирование: объектная декомпозиция; достоинства и недостатки.
5. Абстрагирование.
6. Инкапсуляция (ограничение доступа).
7. Наследование (иерархичность).
8. Полиморфизм (типизация).
9. Анализ задачи.
10. Объектная декомпозиция.
11. Логическое проектирование.
12. Физическое проектирование.
13. Эволюция системы.
14. Модификация проекта.

Тема 2 Техника объектно-ориентированного программирования

1. Статические и экземплярные ресурсы.
2. Поля и методы класса.
3. Конструкторы и деструкторы.
4. Композиция, контейнерные классы.
5. Наследование: родители и потомки.

6. Полиморфизм: раннее и позднее связывание.
7. Полиморфизм на основе абстрактных классов.
8. Полиморфизм на основе виртуальных методов.
9. Перегрузка методов.
10. Перегрузка операций.
11. Делегирование методов, статическое и динамическое делегирование.
12. Параметризованные классы.
13. Интерфейсы.
14. Генерация исключений.
15. Порядок обработки исключений.

### **Повышенный уровень**

Тема 1 Теоретические основы объектно-ориентированного программирования

1. Модульность.
2. Параллелизм.
3. Устойчивость.
4. Структура класса.
5. Статические и динамические ресурсы.

Тема 2 Техника объектно-ориентированного программирования

1. Модификаторы доступа.
2. Свойства и индексы, стратегии доступа.
3. Простое и множественное наследование.
4. Таблица виртуальных методов.
5. Делегирование методов, статическое и динамическое делегирование.
6. Интерфейсы.
7. Обработывающая конструкция `try ... catch`.
8. Финализирующая конструкция `try ... finally`.

### **Вопросы к экзамену**

#### **Пороговый уровень**

1. Эволюция технологии разработки программных продуктов
2. Основные принципы объектно-ориентированного программирования
3. Классы и объекты
4. Организация класса. Модификаторы доступа. Статические и экземплярные ресурсы.
5. Поля и методы класса.
6. Конструкторы и деструкторы.
7. Генерация исключений. Порядок обработки исключений.

8. Обработка конструкция try ... catch.
9. Финализирующая конструкция try ... finally.

### **Повышенный уровень**

1. Объектная декомпозиция
2. Разработка программ с использованием объектно-ориентированной технологии
3. Композиция. Контейнерные классы.
4. Наследование: родители и потомки.
5. Простое и множественное наследование.
6. Полиморфизм: раннее и позднее связывание.
7. Полиморфизм на основе абстрактных классов.
8. Полиморфизм на основе виртуальных методов.
9. Статическое делегирование методов.
10. Динамическое делегирование методов.
11. Параметризованные классы.
12. Интерфейсы.

### **Компетентностно-ориентированные задания и задачи**

#### **ЗАДАЧА 1**

**Цель:**

Ознакомление с концепциями инкапсуляции и модульности. Изучение приемов работы с классами, конструкторами и деструкторами, разработка интерфейса методов класса, создание и работа с экземплярами класса. Освоение принципа «класс-элемент — класс-набор».

**Задание:**

Дан объект согласно вариантам, который является элементом для набора. Элемент состоит из компонент, которые хранятся в нем. Над элементом определены операции:

- получение значения компоненты элемента;
- установка и инициализация значения компоненты элемента;
- контроль значения компоненты элемента (на допустимый диапазон);
- копирование элемента.

Из элементов строится набор. Элементы в наборе проиндексированы от стартового значения. Размер набора задается при создании. Над набором определены операции:

- установка стартового индекса, получение диапазона индексов;
- заполнение набора случайными значениями;
- получение и изменение элемента набора по индексу;
- сортировка элементов по возрастанию и по убыванию;
- дополнительные операции согласно вариантам.

Необходимо разработать:

- класс для описания элемента и его свойств;
- класс для описания набора и его свойств;
- методы работы с элементом и с набором для перечисленных операций;
- дефолтный, копирующий, параметрический конструкторы для создания экземпляров набора и экземпляров элемента;
- интерфейс для редактирования элемента и интерфейс для редактирования набора, отображения и изменения их свойств.

### **Требования:**

Интерфейс и все классы реализуются в одном модуле. Для редактирования элемента разработать функцию `ModifyElement()`, которая должна получать ссылку на экземпляр элемента и предоставлять интерактивный консольный интерфейс для работы с ним. Для редактирования набора разработать функцию `ModifyPalette()`, которая должна получать ссылку на экземпляр набора и предоставлять интерактивный консольный интерфейс для работы с ним. Функция `Main()` запрашивает количество элементов, создает экземпляр набора параметрическим конструктором и вызывает функцию `ModifyPalette()`, которая использует `ModifyElement()`.

### **Комментарии:**

Набор хранит элементы как динамический массив, определяя его размер при создании. При копировании набора, копируются все его элементы.

### Варианты заданий:

0. Элемент: цвет в формате C<sub>Y</sub>M<sub>K</sub>. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.
1. Элемент: положение солнца в координатах  $\alpha$ -азимут,  $z$ -зенит,  $\wedge$ -горизонт. Дополнительно: сортировка по компонентам и целиком, пересчет в другую систему координат (на выбор).
2. Элемент: цвет в формате YUV. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.
3. Элемент: положение солнца в экваториальных координатах  $\square\square$ -склонение,  $p$  — полярное расстояние,  $t$  — часовой угол. Дополнительно: сортировка по компонентам и целиком, пересчет в другую систему координат (на выбор).
4. Элемент: цвет в формате AHSL. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.
5. Элемент: цвет в формате RYB. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.
6. Элемент: декартовы координаты в пространстве  $(x, y)$  — координаты. Дополнительно: сортировка по компонентам и целиком, пересчет в цилиндрическую систему координат.
7. Элемент: цвет в формате YIQ. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.
8. Элемент: время в формате  $h$  — часы,  $m$  — минуты,  $s$  — секунды. Дополнительно: сортировка по компонентам и целиком, пересчет в 12-ти часовой формат времени (AM и PM).
9. Элемент: цвет в формате HSV. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.

## ЗАДАЧА 2

**Цель:**

Ознакомление с концепциями полиморфизма и типизации. Изучение перегрузки операторов. Разработка классов с перегруженными операторами и программирование выражений с их помощью.

**Задание:**

Дан объект данных, над которым определены операции согласно вариантам. Реализовать набор операций для работы с объектом так, чтобы его можно было использовать в выражениях, не прибегая к вызову функций. Необходимо разработать:

- класс объекта и определить правила выполнения операций над ним;
- набор перегруженных операторов, реализующих операции с объектом;
- интерфейс для редактирования объекта с помощью операторов;
- интерфейс для тестирования использования объекта в выражениях.

**Требования:**

Интерфейс и класс объекта реализуются в одном модуле. Для редактирования объекта разработать функцию `ModifyObject()`, которая должна получать ссылку на экземпляр объекта и предоставлять интерактивный консольный интерфейс для работы с ним. Для тестирования использования объекта в выражениях разработать функцию `Main()`, которая должна предоставлять интерактивный консольный интерфейс, демонстрирующий применение объекта в выражениях.

Обязательной реализации подлежат следующие операции: сложение (+ и +=), вычитание (- и -=), умножение (\* и \*=), сравнение на равенство (== и !=), унарные (+, -), инверсия (~), присваивание (=), проверка на ноль (!), преобразование к типу (type), ввод из потока (cin>>) и вывод в поток (cout<<). Разра-

ботать набор тестовых примеров, демонстрирующих использование перегруженных операторов в арифметических и логических выражениях.

### Комментарии:

При перегрузке операторов необходимо учитывать:

- нельзя определить новый лексический символ для оператора;
- нельзя изменить приоритет операторов;
- нельзя изменить арифметичность операторов;
- нельзя перегрузить оператор для стандартных типов данных;
- нельзя перегружать не перегружаемые операторы (`.`), (`::`), (`?_:_`), (`sizeof`), (`typeid`), (`,`);
- некоторые операторы можно перегружать только как методы класса (`=`), (`[ ]`), (`()`), (`->`);
- если оператор можно использовать и как унарный, и как бинарный, например, (`&`), (`*`), (`+`), (`-`), то каждый способ применения перегружается отдельно;
- перегруженные операторы не могут иметь аргументов по умолчанию;
- перегруженные операторы должны учитывать смысловую эквивалентность: `var = var + 1`; `var += 1`; `var++`; `++var`; `var -= (-1)`;
- поведение перегруженных операторов должно соответствовать их смысловому содержанию для определяемых типов данных.

### Варианты заданий:

0. Объект: комплексное число (вещественная и мнимая части).  
Принять: (!) — проверка на ноль, (+ и +=) — сложение, (- и -=) — вычитание, (\* и \*=) — умножение, (== и !=) — сравнение, (double) — вычисление модуля, (float) — вычисление аргумента, (~) — сопряженное число.

1. Объект: интервал времени (часы, минуты, секунды). Реализовать операции с учетом ограничений на часы (0 до 23), минуты и секунды (0

до 59), т.е. результат всегда от 0:0:0 до 23:59:59. Принять: (+ и +=) — сложение, (- и -=) — вычитание, (\* и \*=) — удлинение или сокращение, (!) — проверка на ноль, (== и !=) — сравнение, (long) — преобразование в секунды, (float) — преобразование в часы (3600 сек), (~) — дополнение до конца суток.

2. Объект: денежная сумма (признак валюты [p., \$], сумма в номинале [рубли, доллары], сумма в размене [копейки, центры]). Реализовать операции с учетом конвертации, если валюты не совпадают. Принять: (+ и +=) — сложение, (- и -=) — вычитание, (\* и \*=) — умножение, (!) — проверка на ноль, (== и !=) — сравнение, (float) — в номинал, (int) — в размен, (~) — изменение признака валюты с конвертацией, (%) — процент от суммы.

3. Объект: интервал даты (часов, дней, лет). Реализовать операции с учетом столетия (0 до 99) и ограничений на дни (0 до 364) и часы (0 до 23), т.е. результат всегда от 0-0-0 до 23-364-99. Принять: (\* и \*=) — удлинение или сокращение, (+ и +=) — сложение, (- и -=) — вычитание, (== и !=) — сравнение, (!) — проверка на ноль, (long) — преобразование в часы, (float) — преобразование в года (365 дней), (~) — дополнение до конца столетия.

4. Объект: расстояние (сажень, аршин, вершок). 1 сажень = 3 аршинам, 1 аршин = 16 вершкам, 1 вершок = 44,5 мм. Результат всегда от 0 до 500 сажений (1 верста). Принять: (+ и +=) — сложение, (- и -=) — разность, (\* и \*=) — удлинение или сокращение, (== и !=) — сравнение, (!) — проверка на ноль, (double) — преобразование в миллиметры, (int) — преобразование в вершки, (~) — дополнение до версты (500 сажений).

5. Объект: строка символов (0 до 128). Принять: (+ и +=) — соединение строк, повторение символа, (- и -=) — отсечение строки, (\*) — поиск подстроки, (\*=) — заполнение подстрокой или символом, (== и !=) — сравнение, (!) — проверка на пусто, (~) — переворот наоборот,

(int) — длина строки.

6. Объект: натуральная дробь (целое, числитель, знаменатель). Реализовать операции с учетом приведения к общему знаменателю. Принять: (+ и +=) — сложение, (- и -=) — вычитание, (\* и \*=) — умножение, (!) — проверка на ноль, (== и !=) — сравнение, (double) — преобразование в рациональную дробь, (~) — взаимнообратная натуральная дробь.

7. Объект: угол (градусы, минуты, секунды). Реализовать операции с учетом целых оборотов и ограничений на градусы (0 до 359), минуты и секунды (0 до 59), т.е. результат всегда от 0°0'0" до 359°59'59". Принять: (+ и +=) — сложение, (- и -=) — вычитание, (\* и \*=) — умножение, (!) — проверка на ноль, (== и !=) — сравнение, (double) — преобразование в радианы, (int) — преобразование в секунды, (~) — обратный угол до 360°.

8. Объект: квадратная матрица [3x3]. Реализовать операции над матрицами. Принять: (+ и +=) — сложение, (- и -=) — вычитание, (\* и \*=) — умножение, (!) — проверка на ноль, (== и !=) — сравнение, (double) — вычисление детерминанта, (int) — количество ячеек, (~) — транспонирование.

9. Объект: алфавит (только прописные от A до Z). Реализовать операции над алфавитами как над множествами. Принять: (+ и +=) — объединение, (!) — проверка на пусто, (- и -=) — разность, (\* и \*=) — пересечение, (== и !=) — сравнение, (int) — количество букв, (~) — отрицание как замена на буквы, которых нет.

### ЗАДАЧА 3

#### Цель:

Ознакомление с концепциями наследования и абстракции. Обычное и множественное наследование классов. Статические свойства, методы, классы. Перегрузка методов при наследовании классов. Освоение приемов создания,

редактирования, копирования, удаления экземпляров и разработки интерфейсов классов.

**Задание:**

Дана фигура на плоскости согласно вариантам. Фигура описывается индивидуальными геометрическими свойствами и общими оформительскими свойствами: цвет, видимость. У фигуры имеются характеристики: периметр, площадь, ограничивающая область. Область размещения фигур в плоскости ограничена экстендами, за которые фигура не должна выходить.

Необходимо разработать:

- классы для описания положения «Location» и ограничивающей области «Clip» в плоскости;
- статический класс «Geometry» для хранения общих констант и методов проверки различных ограничений на размещение фигур в плоскости;
- класс геометрического примитива «Primitive» для хранения и редактирования оформительских свойств фигуры как наследника от статического класса «Geometry»;
- класс примитивной фигуры — точки «Point» как наследника от классов «Location» и «Primitive»;
- класс фигуры согласно варианту «Figure» как наследника от класса «Point» с описанием специфических свойств и методов фигуры;
- наборы конструкторов для создания экземпляров каждого класса различными способами (дефолтный, копирующий, параметрический);
- методы для изменения свойств и вычисления характеристик фигуры;
- интерфейс для отображения и изменения всех свойств фигуры.

**Требования:**

Интерфейс и классы реализуются в одном модуле. Для редактирования фигуры разработать функцию `ModifyFigure()`, которая должна получать ссылку на

экземпляр фигуры и предоставлять интерактивный консольный интерфейс для работы с ним.

### **Варианты заданий:**

0. Фигура: сектор окружности.
1. Фигура: треугольник Рело.
2. Фигура: правильный шестиугольник.
3. Фигура: эллипс.
4. Фигура: параллелограмм.
5. Фигура: сегмент окружности.
6. Фигура: ромб.
7. Фигура: кольцо (бублик).
8. Фигура: правильная трапеция.
9. Фигура: дельтоид.

## **5. ПОДГОТОВКА К ЗАНЯТИЯМ**

### **5.1. Подготовка к лекциям**

Главное в период подготовки к лекционным занятиям — научиться методам самостоятельного умственного труда, сознательно развивать свои творческие способности и овладевать навыками творческой работы. Для этого необходимо строго соблюдать дисциплину учебы и поведения. Четкое планирование своего рабочего времени и отдыха является необходимым условием для успешной самостоятельной работы. В основу его нужно положить рабочие программы изучаемых в семестре дисциплин.

Каждому студенту следует составлять еженедельный и семестровый планы работы, а также план на каждый рабочий день. С вечера всегда надо распределять работу на завтрашний день. В конце каждого дня целесообразно подводить итог работы: тщательно проверить, все ли выполнено по намеченному плану, не было ли каких-либо отступлений, а если были, по какой причине это произошло. Нужно осуществлять самоконтроль, который явля-

ется необходимым условием успешной учебы. Если что-то осталось невыполненным, необходимо изыскать время для завершения этой части работы, не уменьшая объема недельного плана.

Слушание и запись лекций — сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность студента. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим студентом. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Запись лекций рекомендуется вести по возможности собственными формулировками. Желательно запись осуществлять на одной странице, а следующую оставлять для проработки учебного материала самостоятельно в домашних условиях.

Конспект лекций лучше подразделять на пункты, параграфы, соблюдая красную строку. Этому в большой степени будут способствовать пункты плана лекции, предложенные преподавателям. Принципиальные места, определения, формулы и другое следует сопровождать замечаниями «важно», «особо важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек. Лучше если они будут собственными, чтобы не приходилось просить их у однокурсников и тем самым не отвлекать их во время лекции. Целесообразно разработать собственную «маркографию» (значки, символы), сокращения слов. Не лишним будет и изучение основ стенографии. Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть знаниями.

## 5.2. Подготовка к практическим занятиям

Подготовку к каждому практическому занятию студент должен начать с ознакомления с методическими указаниями, которые включают содержание работы. Тщательное продумывание и изучение вопросов основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованную к данной теме. На основе индивидуальных предпочтений студенту необходимо самостоятельно выбрать тему доклада по проблеме и по возможности подготовить по нему презентацию.

Если программой дисциплины предусмотрено выполнение практического задания, то его необходимо выполнить с учетом предложенной инструкции (устно или письменно). Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы семинара, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий и контрольных работ.

В зависимости от содержания и количества отведенного времени на изучение каждой темы практическое занятие может состоять из четырех-пяти частей:

1. Обсуждение теоретических вопросов, определенных программой дисциплины.
2. Доклад и/или выступление с презентациями по выбранной проблеме.
3. Обсуждение выступлений по теме — дискуссия.
4. Выполнение практического задания с последующим разбором полученных результатов или обсуждение практического задания.
5. Подведение итогов занятия.

Первая часть — обсуждение теоретических вопросов — проводится в виде фронтальной беседы со всей группой и включает выборочную проверку преподавателем теоретических знаний студентов. Примерная продолжительность — до 15 минут. Вторая часть — выступление студентов с докладами, которые должны сопровождаться презентациями с целью усиления наглядности восприятия, по одному из вопросов практического занятия. Обязательный элемент доклада — представление и анализ статистических данных, обоснование социальных последствий любого экономического факта, явления или процесса. Примерная продолжительность — 20-25 минут. После докладов следует их обсуждение — дискуссия. В ходе этого этапа практического занятия могут быть заданы уточняющие вопросы к докладчикам. Примерная продолжительность — до 15-20 минут. Если программой предусмотрено выполнение практического задания в рамках конкретной темы, то преподавателями определяется его содержание и дается время на его выполнение, а затем идет обсуждение результатов. Подведением итогов заканчивается практическое занятие.

В процессе подготовки к практическим занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме семинарского или практического занятия, что позволяет студентам проявить свою индивидуальность в рамках выступления на данных занятиях, выявить широкий спектр мнений по изучаемой проблеме.

### 5.3. Самостоятельное изучение материала тем

Конспект — наиболее совершенная и наиболее сложная форма записи. Слово «конспект» происходит от латинского «conspectus», что означает «обзор, изложение». В правильно составленном конспекте обычно выделено самое основное в изучаемом тексте, сосредоточено внимание на наиболее существенном, в кратких и четких формулировках обобщены важные теоретические положения.

Конспект представляет собой относительно подробное, последовательное изложение содержания прочитанного. На первых порах целесообразно в записях ближе держаться тексту, прибегая зачастую к прямому цитированию автора. В дальнейшем, по мере выработки навыков конспектирования, записи будут носить более свободный и сжатый характер.

Конспект книги обычно ведется в тетради. В самом начале конспекта указывается фамилия автора, полное название произведения, издательство, год и место издания. При цитировании обязательная ссылка на страницу книги. Если цитата взята из собрания сочинений, то необходимо указать соответствующий том. Следует помнить, что четкая ссылка на источник — неперемutable правило конспектирования. Если конспектируется статья, то указывается, где и когда она была напечатана.

Конспект подразделяется на части в соответствии с заранее продуманным планом. Пункты плана записываются в тексте или на полях конспекта. Писать его рекомендуется четко и разборчиво, так как небрежная запись с течением времени становится малопонятной для ее автора. Существует правило: конспект, составленный для себя, должен быть по возможности написан так, чтобы его легко прочитал и кто-либо другой.

Формы конспекта могут быть разными и зависят от его целевого назначения (изучение материала в целом или под определенным углом зрения, подготовка к докладу, выступлению на занятии и т.д.), а также от характера

произведения (монография, статья, документ и т.п.). Если речь идет просто об изложении содержания работы, текст конспекта может быть сплошным, с выделением особо важных положений подчеркиванием или различными значками.

В случае, когда не ограничиваются переложением содержания, а фиксируют в конспекте и свои собственные суждения по данному вопросу или дополняют конспект соответствующими материалами их других источников, следует отводить место для такого рода записей. Рекомендуется разделить страницы тетради пополам по вертикали и в левой части вести конспект произведения, а в правой свои дополнительные записи, совмещая их по содержанию.

Конспектирование в большей мере, чем другие виды записей, помогает вырабатывать навыки правильного изложения в письменной форме важные теоретических и практических вопросов, умение четко их формулировать и ясно излагать своими словами.

Таким образом, составление конспекта требует вдумчивой работы, затраты времени и труда. Зато во время конспектирования приобретаются знания, создается фонд записей.

Конспект может быть текстуальным или тематическим. В текстуальном конспекте сохраняется логика и структура изучаемого произведения, а запись ведется в соответствии с расположением материала в книге. За основу тематического конспекта берется не план произведения, а содержание какой-либо темы или проблемы.

Текстуальный конспект желательно начинать после того, как вся книга прочитана и продумана, но это, к сожалению, не всегда возможно. В первую очередь необходимо составить план произведения письменно или мысленно, поскольку в соответствии с этим планом строится дальнейшая работа. Конспект включает в себя тезисы, которые составляют его основу. Но, в отличие от тезисов, конспект содержит краткую запись не только выводов, но и доказательств, вплоть до фактического материала. Иначе говоря, конспект — это

расширенные тезисы, дополненные рассуждениями и доказательствами, мыслями и соображениями составителя записи.

Как правило, конспект включает в себя и выписки, но в него могут войти отдельные места, цитируемые дословно, а также факты, примеры, цифры, таблицы и схемы, взятые из книги. Следует помнить, что работа над конспектом только тогда будет творческой, когда она не ограничена текстом изучаемого произведения. Нужно дополнять конспект данными из другими источниками.

В конспекте необходимо выделять отдельные места текста в зависимости от их значимости. Можно пользоваться различными способами: подчеркиваниями, вопросительными и восклицательными знаками, репликами, краткими оценками, писать на полях своих конспектов слова: «важно», «очень важно», «верно», «характерно».

В конспект могут помещаться диаграммы, схемы, таблицы, которые придадут ему наглядность.

Составлению тематического конспекта предшествует тщательное изучение всей литературы, подобранной для раскрытия данной темы. Бывает, что какая-либо тема рассматривается в нескольких главах или в разных местах книги. А в конспекте весь материал, относящийся к теме, будет сосредоточен в одном месте. В плане конспекта рекомендуется делать пометки, к каким источникам (вплоть до страницы) придется обратиться для раскрытия вопросов. Тематический конспект составляется обычно для того, чтобы глубже изучить определенный вопрос, подготовиться к докладу, лекции или выступлению на семинарском занятии. Такой конспект по содержанию приближается к реферату, докладу по избранной теме, особенно если включает и собственный вклад в изучение проблемы.

## 5.4. Подготовка к экзамену

Экзаменационная сессия — очень тяжелый период работы для студентов и ответственный труд для преподавателей. Главная задача экзаменов — проверка качества усвоения содержания дисциплины.

На основе такой проверки оценивается учебная работа не только студентов, но и преподавателей: по результатам экзаменов можно судить и о качестве всего учебного процесса. При подготовке к экзамену студенты повторяют материал курсов, которые они слушали и изучали в течение семестра, обобщают полученные знания, выделяют главное в предмете, воспроизводят общую картину для того, чтобы яснее понять связь между отдельными элементами дисциплины.

При подготовке к экзаменам основное направление дают программы курса и конспект, которые указывают, что в курсе наиболее важно. Основной материал должен прорабатываться по учебнику, поскольку конспекта недостаточно для изучения дисциплины. Учебник должен быть проработан в течение семестра, а перед экзаменом важно сосредоточить внимание на основных, наиболее сложных разделах. Подготовку по каждому разделу следует заканчивать восстановлением в памяти его краткого содержания в логической последовательности.

До экзамена обычно проводится консультация, но она не может возместить отсутствия систематической работы в течение семестра и помочь за несколько часов освоить материал, требующийся к экзамену. На консультации студент получает лишь ответы на трудные или оставшиеся неясными вопросы. Польза от консультации будет только в том случае, если студент до нее проработает весь материал. Надо учиться задавать вопросы, вырабатывать привычку пользоваться справочниками, энциклопедиями, а не быть на иждивении у преподавателей, который не всегда может тут же, «с ходу» назвать какой-либо факт, имя, событие. На экзамене нужно показать не только знание предмета, но и умение логически связно построить устный ответ.

Получив билет, надо вдуматься в поставленные вопросы для того, чтобы правильно понять их. Нередко студент отвечает не на тот вопрос, который поставлен, или в простом вопросе ищет скрытого смысла. Не поняв вопроса и не обдумав план ответа, не следует начинать писать. Конспект своего ответа надо рассматривать как план краткого сообщения на данную тему и составлять ответ нужно кратко. При этом необходимо показать умение выражать мысль четко и доходчиво.

Отвечать нужно спокойно, четко, продуманно, без торопливости, придерживаясь записи своего ответа. На экзаменах студент показывает не только свои знания, но и учится владеть собой. После ответа на билет могут следовать вопросы, которые имеют целью выяснить понимание других разделов курса, не вошедших в билет. Как правило, на них можно ответить кратко, достаточно показать знание сути вопроса. Часто студенты при ответе на дополнительные вопросы проявляют поспешность: не поняв смысла того, что у них спрашивают, начинают отвечать и нередко говорят не по сути.

Следует помнить, что необходимым условием правильного режима работы в период экзаменационной сессии является нормальный сон, поэтому подготовка к экзаменам не должна быть в ущерб сну. Установлено, что сильное эмоциональное напряжение во время экзаменов неблагоприятно отражается на нервной системе и многие студенты из-за волнений не спят ночи перед экзаменами. Обычно в сессию студенту не до болезни, так как весь организм озабочен одним — сдать экзамены. Но это еще не значит, что последствия неправильно организованного труда и чрезмерной занятости не скажутся потом. Поэтому каждый студент помнить о важности рационального распорядка рабочего дня и о своевременности снятия или уменьшения умственного напряжения.

## 6. ПЛАН-ГРАФИК ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

На первом этапе необходимо ознакомиться с рабочей программой дисциплины, в которой рассмотрено содержание тем дисциплины лекционного курса, взаимосвязь тем лекций с лабораторными и практическими занятиями, темы и виды самостоятельной работы. По каждому виду самостоятельной работы предусмотрены определённые формы отчетности.

### Технологическая карта самостоятельной работы студента очной формы обучения

Коды реализуемых компетенций	Вид деятельности студентов	Итоговый продукт самостоятельной работы	Средства и технологии оценки	Объем часов, в том числе		
				СРС	Контактная работа с преподавателем	Всего
<b>6 семестр</b>						
ПК-2	Подготовка к лекции	Самостоятельно изученный теоретический материал,	Собеседование	2,28	0,12	2,40
ПК-2	Подготовка к практическому занятию	Самостоятельно изученный теоретический материал,	Собеседование	4,56	0,24	4,80
ПК-2	Самостоятельное изучение литературы	Самостоятельно изученный теоретический материал,	Собеседование	56,57	2,98	59,55
ПК-2	Подготовка к экзамену	Самостоятельно изученный теоретический материал,	Устный экзамен	18,75	1,50	20,25

<b>Итого за семестр</b>	82,16	4,84	87,00
<b>Итого</b>	82,16	4,84	87,00

**Технологическая карта самостоятельной работы студента заочной формы обучения**

Коды реализуемых компетенций	Вид деятельности студентов	Итоговый продукт самостоятельной работы	Средства и технологии оценки	Объем часов, в том числе		
				СРС	Контактная работа с преподавателем	Всего
<b>3 курс</b>						
ПК-2	Подготовка к лекции	Самостоятельно изученный теоретический материал,	Собеседование	0,43	0,02	0,45
ПК-2	Подготовка к практическому занятию	Самостоятельно изученный теоретический материал,	Собеседование	0,85	0,05	0,90
ПК-2	Самостоятельное изучение литературы	Самостоятельно изученный теоретический материал,	Собеседование	112,00	5,90	117,90
ПК-2	Подготовка к экзамену	Самостоятельно изученный теоретический материал,	Устный экзамен	5,25	1,50	6,75
<b>Итого за курс</b>				118,53	7,47	126,00
<b>Итого</b>				118,53	7,47	126,00

## 7. КОНТРОЛЬНЫЕ ТОЧКИ И ОТЧЕТНОСТЬ ПО НИМ

В рамках рейтинговой системы успеваемость студентов по дисциплине оценивается в ходе текущего контроля успеваемости и промежуточной аттестации.

### Текущий контроль

№ п/п	Вид деятельности студентов	Сроки выполнения	Количество баллов
<b>6 семестр</b>			
1	Практическое занятие 11	11	15
2	Практическое занятие 15	15	15
3	Практическое занятие 16	16	25
	<b>Итого за семестр:</b>		<b>55</b>
	<b>Итого:</b>		<b>55</b>

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

### Промежуточная аттестация

Промежуточная аттестация в форме экзамена предусматривает проведение обязательной экзаменационной процедуры и оценивается 40 баллами из 100. Минимальное количество баллов, необходимое для допуска к экзамену, составляет 33 балла. Положительный ответ студента на экзамене оценивается

рейтинговыми баллами в диапазоне от **20** до **40** ( $20 \leq S_{\text{экз}} \leq 40$ ), оценка меньше 20 баллов считается неудовлетворительной.

Шкала соответствия рейтингового балла экзамена 5-балльной системе

Рейтинговый балл по дисциплине	Оценка по 5-балльной системе
<b>35-40</b>	Отлично
<b>28-34</b>	Хорошо
<b>20-27</b>	Удовлетворительно

Итоговая оценка по дисциплине, изучаемой в одном семестре, определяется по сумме баллов, набранных за работу в течение семестра, и баллов, полученных при сдаче экзамена:

Шкала пересчета рейтингового балла по дисциплине в оценку по 5-балльной системе

Рейтинговый балл по дисциплине	Оценка по 5-балльной системе
<b>88-100</b>	Отлично
<b>72-87</b>	Хорошо
<b>53-71</b>	Удовлетворительно
<b>менее 53</b>	Неудовлетворительно

Для студентов заочной формы обучения рейтинговая оценка знаний не предусмотрена.

## ЛИТЕРАТУРА

### Перечень основной литературы:

1. Иванова Г.С. Объектно-ориентированное программирование [Электронный ресурс]: учебник/ Иванова Г.С., Ничушкина Т.Н. — Электрон. текстовые данные. — Москва: Московский государственный технический университет имени Н.Э. Баумана, 2014. — 456 с. — Режим доступа: <http://www.iprbookshop.ru/94030.html>. — ЭБС «IPRbooks».
2. Маляров А.Н. Объектно-ориентированное программирование [Электронный ресурс]: учебник для технических вузов/ Маляров А.Н. — Электрон. текстовые данные. — Самара: Самарский государственный технический университет, ЭБС АСВ, 2017. — 332 с. — Режим доступа: <http://www.iprbookshop.ru/91772.html>. — ЭБС «IPRbooks».
3. Мурадханов С.Э. Информатика и программирование: объектно-ориентированное программирование (на основе языка С#) [Электронный ресурс]: учебник/ Мурадханов С.Э., Широков А.И. — Электрон. текстовые данные. — Москва: Издательский Дом МИСиС, 2015. — 309 с. — Режим доступа: <http://www.iprbookshop.ru/98855.html>. — ЭБС «IPRbooks».

### Перечень дополнительной литературы:

1. Зыков С.В. Введение в теорию программирования. Объектно-ориентированный подход [Электронный ресурс]: учебное пособие/ Зыков С.В. — Электрон. текстовые данные. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 187 с. — Режим доступа: <http://www.iprbookshop.ru/102007.html>. — ЭБС «IPRbooks».
2. Николаев Е.И. Объектно-ориентированное программирование [Электронный ресурс]: учебное пособие/ Николаев Е.И. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2015. — 225 с. — Режим доступа: <http://www.iprbookshop.ru/62967.html>. — ЭБС «IPRbooks».

3. Сорокин А.А. Объектно-ориентированное программирование [Электронный ресурс]: учебное пособие. Курс лекций/ Сорокин А.А. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2014. — 174 с. — Режим доступа:

<http://www.iprbookshop.ru/63110.html>. — ЭБС «IPRbooks».