

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Воронежский государственный технический университет»

УТВЕРЖДАЮ  
Декан факультета экономики, менеджмента и  
информационных технологий



/ Баркалов С.А./

31 августа 2021 г.

**РАБОЧАЯ ПРОГРАММА**  
дисциплины  
**«Объектно-ориентированное программирование»**

Направление подготовки 09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

Профиль «Прикладная информатика в экономике цифрового общества»

Квалификация выпускника бакалавр

Нормативный период обучения 4 года/ 4 года 11 месяцев

Форма обучения очная/заочная

Год начала подготовки 2021

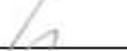
Автор программы

 / Акамсина Н.В./

И.о. заведующего кафедрой  
систем управления и  
информационных  
технологий в строительстве

 / Десятирикова Е.Н./

Руководитель ОПОП

 / Аснина Н.Г./

Воронеж 2021

## 1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

### 1.1. Цели дисциплины

формирование профессиональных компетенций по проектированию и сопровождению и эксплуатации программных систем на основе объектно-ориентированного моделирования предметной области

### 1.2. Задачи освоения дисциплины

изучение теоретических основ объектно-ориентированного программирования на примере языка высокого уровня;

получение практических навыков программирования и применения объектно-ориентированного подхода к решению инженерных задач.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Объектно-ориентированное программирование» относится к дисциплинам обязательной части блока Б1.

## 3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Объектно-ориентированное программирование» направлен на формирование следующих компетенций:

ОПК-2 - Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;

ОПК-4 - Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью;

ОПК-6 - Способен анализировать и разрабатывать организационно-технические и экономические процессы с применением методов системного анализа и математического моделирования;

ОПК-7 - Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-2	знать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности
	уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности
	владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности

ОПК-4	знать основные этапы и принципы создания программ
	уметь разрабатывать, согласовывать и выпускать все виды документации программных средств и оценивать качество ПО
ОПК-6	владеть методами и средствами представления данных, техниками извлечения требований и знаний о предметной области
	знать абстракции, рекурсии, типизации, повторного использования, обработки исключений
	уметь разрабатывать программы с графическим интерфейсом пользователя на объектно-ориентированном языке программирования
ОПК-7	владеть навыками работы с инструментальными средами разработки программного обеспечения
	знать основы теории систем и системного анализа, моделирования информационных систем
	уметь применять методы моделирования для автоматизации задач принятия решений, анализа информационных потоков, расчета экономической эффективности и надежности информационных систем и технологий.
ОПК-7	владеть навыками проведения инженерных расчетов основных показателей результативности создания и применения информационных систем и технологий

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Объектно-ориентированное программирование» составляет 5 з.е.

Распределение трудоемкости дисциплины по видам занятий  
**очная форма обучения**

Виды учебной работы	Всего часов	Семестры	
		4	5
<b>Аудиторные занятия (всего)</b>	126	54	72
В том числе:			
Лекции	36	18	18
Практические занятия (ПЗ)	18	-	18
Лабораторные работы (ЛР)	72	36	36
<b>Самостоятельная работа</b>	27	18	9
<b>Курсовой проект</b>	+		+
Часы на контроль	27	-	27
Виды промежуточной аттестации - экзамен, зачет с оценкой	+	+	+
Общая трудоемкость:			

академические часы	180	72	108
зач.ед.	5	2	3

### заочная форма обучения

Виды учебной работы	Всего часов	Семестры		
		2	3	4
<b>Аудиторные занятия (всего)</b>	40	-	10	10
В том числе:				
Лекции	12	-	4	2
Практические занятия (ПЗ)	4	-	-	2
Лабораторные работы (ЛР)	24	-	6	6
<b>Самостоятельная работа</b>	294	-	58	89
<b>Курсовой проект</b>	+			+
Часы на контроль	26	-	4	9
Виды промежуточной аттестации - экзамен, зачет с оценкой	+		+	+
Общая трудоемкость:				
академические часы	180	0	72	108
зач.ед.	5	0	2	3

## 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

#### очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Прак зан.	Лаб. зан.	СРС	Всего, час
1	Основы объектно-ориентированной парадигмы	Понятие объектно-ориентированной методологии программирования (ООП). Сравнение концепции объектноориентированного программирования и структурной. Абстракция типов данных. Язык UML. Основные принципы ООП.	6	2	12	4	24
2	Абстракция и инкапсуляция	Абстрактный тип данных. Значение типизации в программировании. Состояние и поведение объектов.. Понятие класса, его структура. Экземпляры класса. Создание, копирование и удаление объектов. Управление объектами и доступом. Интерфейс. Квантор видимости.	8	4	14	5	31
3	Наследование	Основы наследования. Конструкторы и порядок их вызова. Перегрузка методов. Присваивание и сравнение объектов. Динамическая диспетчеризация методов. Перегрузка операторов. Одиночное наследование классов. Типизация и наследование. Иерархическое упорядочивание в программе. Полиморфизм. Создание многоуровневой иерархии. Абстрактные и виртуальные методы. Интерфейсы и абстрактные классы. Обобщение.	8	4	16	6	34

4	Принципы объектно-ориентированного проектирования	Агрегация как альтернатива наследования. Принципы SOLID. Рассмотрение принципов объектно-ориентированного программирования на примере языка Java. Паттерны проектирования. Основы обработки исключений. Создание собственных подклассов исключений.	8	4	14	6	32
5	Методы и средства программирования	Пакеты. Доступ к пакетам и его компонентам. Импорт пакетов. Интерфейсы: объявление, реализация. Вложенные интерфейсы. Статические и закрытые методы в интерфейсах API и фреймворки.. Обработка событий. Работа с элементами управления графического интерфейса.	6	4	16	6	32
<b>Итого</b>			<b>36</b>	<b>18</b>	<b>72</b>	<b>27</b>	<b>153</b>

### заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час	
1	Основы объектно-ориентированной парадигмы	Понятие объектно-ориентированной методологии программирования (ООП). Сравнение концепции объектноориентированного программирования и структурной. Абстракция типов данных. Язык UML. Основные принципы ООП.	2	4	56	68	
2	Абстракция и инкапсуляция	Абстрактный тип данных. Значение типизации в программировании. Состояние и поведение объектов.. Понятие класса, его структура. Экземпляры класса. Создание, копирование и удаление объектов. Управление объектами и доступом. Интерфейс. Квантор видимости.	2	6	58	68	
3	Наследование	Основы наследования. Конструкторы и порядок их вызова. Перегрузка методов. Присваивание и сравнение объектов. Динамическая диспетчеризация методов. Перегрузка операторов. Одиночное наследование классов. Типизация и наследование. Иерархическое упорядочивание в программе. Полиморфизм. Создание многоуровневой иерархии. Абстрактные и виртуальные методы. Интерфейсы и абстрактные классы. Обобщение.	4	4	58	66	
4	Принципы объектно-ориентированного проектирования	Агрегация как альтернатива наследования. Принципы SOLID. Рассмотрение принципов объектно-ориентированного программирования на примере языка Java. Паттерны проектирования. Основы обработки исключений. Создание собственных подклассов исключений.	2	6	62	66	
5	Методы и средства программирования	Пакеты. Доступ к пакетам и его компонентам. Импорт пакетов. Интерфейсы: объявление, реализация. Вложенные интерфейсы. Статические и закрытые методы в интерфейсах API и фреймворки.. Обработка событий. Работа с элементами управления графического интерфейса.	2	4	60	66	
<b>Итого</b>				<b>12</b>	<b>24</b>	<b>294</b>	<b>334</b>

## **5.2 Перечень лабораторных работ**

Знакомство со средой разработки и разработка консольного приложения небольшого размера.

Создание простого GUI приложения.

Абстрактный суперкласса и реализация наследования.

Реализация приложения для рисования фигур.

Реализация обработчиков событий мыши.

Создание приложения с использованием фабричных методов и интерфейсов.

Разработка приложения рисования графических объектов многоуровневой иерархии наследования.

Работа с массивами. Создание классов-адаптеров для работы с различными структурами данных.

Реализация паттерна Декоратор.

Использование паттерна Стратегия в разработке программы.

## **6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ**

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсовых проектов в 4, 3 семестрах для заочной формы обучения.

Примерная тематика курсового проекта:

«Построение и реализации системы каталогизации книг на основе объектно-ориентированного подхода»

«Объектно-ориентированное программирование астрономической модели солнечной системы»

«Построение модели информационно-поисковой системы товаров на складе»

«Объектно-ориентированное программирование модели оптимизации расходов на электроэнергию»

«Объектное моделирование и реализация прототипа системы планирования семейного бюджета»

«Объектно-ориентированное проектирование системы анализа расходов на семейное торжество»

Задачи, решаемые при выполнении курсового проекта:

- разработка модели классов и диаграмм взаимодействия на языке UML;

- реализацию программы с графическим интерфейсом на языке java;

- создание описания пакетов, классов и методов разработанной программы;

- получение практического опыта работы с инструментальными средствами разработки

Курсовой проект включают в себя программный код, размещенный в общедоступном репозитории и пояснительную записку.

## 7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

### 7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

#### 7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ОПК-2	знать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	устный опрос	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	выполнение стандартных практических задач	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	реализация программного продукта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-4	знать основные этапы и принципы создания программ	устный опрос	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь разрабатывать, согласовывать и выпускать все виды документации программных средств и оценивать качество ПО	выполнение стандартных практических задач	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	владеть методами и средствами представления данных, техниками извлечения требований и знаний о предметной области	реализация программного продукта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-6	знать абстракции, рекурсии, типизации, повторного использования, обработки исключений	устный опрос	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь разрабатывать программы с графическим интерфейсом пользователя на объектно-ориентированном языке программирования	выполнение стандартных практических задач	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками работы с инструментальными средами разработки программного обеспечения	реализация программного продукта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-7	знать основы теории систем и системного анализа,	устный опрос	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	моделирования информационных систем	выполнение стандартных практических задач	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь применять методы моделирования для автоматизации задач принятия решений, анализа информационных потоков, расчета экономической эффективности и надежности информационных систем и технологий.	реализация программного продукта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

### 7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 2, 3, 3, 4 семестре для очной формы обучения по четырехбалльной системе:

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
-------------	---	---------------------	---------	--------	--------	----------

ОПК-2	знать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-4	знать основные этапы и принципы создания программ	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь разрабатывать, согласовывать и выпускать все виды документации программных средств и оценивать качество ПО	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть методами и средствами представления данных, техниками извлечения требований и знаний о предметной области	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-6	знать абстракции, рекурсии, типизации, повторного использования, обработки исключений	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов

	уметь разрабатывать программы с графическим интерфейсом пользователя на объектно-ориентированном языке программирования	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками работы с инструментальными средами разработки программного обеспечения	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-7	знать основы теории систем и системного анализа,	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	моделирования информационных систем	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	уметь применять методы моделирования для автоматизации задач принятия решений, анализа информационных потоков, расчета экономической эффективности и надежности информационных систем и технологий.	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

## 7.2 Примерный перечень оценочных средств (типичные контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

### 7.2.1 Примерный перечень заданий для подготовки к тестированию

1. *Формализованное описание процесса обработки данных – это ...*  
программа

алгоритм

информационная среда

транслятор

2 *Принцип, который позволяет создавать иерархию классов, начиная с первоначального класса, и включает элементы классов-предков – это ...*

объединение

наследование

*полиморфизм*

*инкапсуляция*

3 Принцип, который задает единый образ действий для различных родственных классов, - это ...

*объединение*

*наследование*

*полиморфизм*

*инкапсуляция*

4 Методика разработки программ, в основе которой лежит понятие объекта, как некоторой структуры, описывающей объект реального мира, его поведение, - это ...

*детализация*

*инкапсуляция*

*моделирование*

*объектно-ориентированное программирование*

5 Обработчики событий создаются на основе

*делегатов*

*классов*

*объектов*

*интерфейсов*

6. Базовый класс для определения типов окон пользовательского интерфейса приложения – это класс .....

*Object*

*Sender*

*Form*

*System*

7. Конструктор класса *Form* может быть записан:

*public Form()*

*public struct Form();*

*public void Form();*

*public class Form*

8. Укажите квантор видимости для общедоступного атрибута класса.

+

-

\*

#

9. Методы класса определяют:

*какие операции можно выполнять с объектами данного класса*

*какие типы значений могут принимать данные-члены класса*

*каким образом реализуется защита данных-членов класса*

*как изменяется сущность*

10. Класс не имеющий экземпляра или объекта называется:

*Абстрактный;*

*Приватный;*

*Внутренний;*

*Анонимный*

### **7.2.2 Примерный перечень заданий для решения стандартных задач**

*Задача 1. Разработать и реализовать класс Student: Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс.*

*Создать массив объектов.*

*Вывести:*

*а) список студентов заданного факультета;*

*б) списки студентов для каждого факультета и курса;*

*в) список студентов, родившихся после заданного года*

*Задача 2. Разработать и реализовать класс Abiturient: Фамилия, Имя, Отчество, Адрес, Баллы по предметам.*

*Создать список объектов.*

*Вывести:*

*а) список абитуриентов, имеющих неудовлетворительные оценки;*

*б) список абитуриентов, сумма баллов у которых не меньше заданной;*

*в) выбрать N абитуриентов, имеющих самую высокую сумму баллов, и список абитуриентов, имеющих полупроходной балл.*

*Задача 3. Разработать и реализовать класс Apartment: Адрес, Этаж, Количество комнат, Площадь.*

*Создать иерархию объектов.*

*Вывести:*

*а) список квартир, имеющих заданное число комнат;*

*б) список квартир, имеющих заданное число комнат и расположенных на этаже, который находится в определенном промежутке;*

*в) список квартир, имеющих площадь, превосходящую заданную.*

*Задача 4. Определить объект файл с поддержкой операций создания, копирования, перемещения, удаления файла, наполнения содержимым. Унаследовать от него класс «Зашифрованный файл» с поддержкой функций шифрации – дешифрации содержимого. В программе продемонстрировать функциональность разработанных классов.*

*Задача 5. Определить класс «Матрица» с возможностью динамического выделения и освобождения памяти, наполнения матрицы, сохранения и чтения из файла. Унаследовать класс «Треугольная матрица» с приведением матрицы к треугольному виду (если возможно) автоматически при каждом изменении значения элемента матрицы. содержимого. В программе продемонстрировать функциональность разработанных классов.*

*Задача 6. Определить класс «Шахматная фигура» и унаследовать от него классы, соответствующие шахматным фигурам «Ферзь», «Пешка», «Конь». Задать позицию и определить список фигур, которые может убить некоторая выбранная. содержимого.*

*Задача 7 Определить абстрактный класс Currency для работы с денежными суммами. Определить в нем методы перевода в рубли и вывода на экран. На его основе реализовать классы Dollar, Euro и Pound (фунт*

стерлингов) с возможностью пересчета в центы и пенсы соответственно и указанием текущего курса.. Создать класс *Purse* (кошелек), содержащий массив объектов этих классов в динамической памяти. Предусмотреть возможность случайного наполнения кошелька купюрами различного типа и подсчета общей суммы кошелька при изменении курса валют. В программе продемонстрировать функциональность разработанных классов.

Задача 8. Определить абстрактный класс *Function* (функция) с виртуальными методами вычисления значения функции  $y = f(x)$  в заданной точке  $x$  и вывода результата на экран, поиска минимума и максимума функции на заданном интервале.. Унаследовать от класса *Function* классы *Hiperbola* и *Parabola*, *Exponenta*, в которых уточняется функция  $f(x)$  и задаются коэффициенты соответствующих функций. Продемонстрировать функциональные возможности этих классов (получением значений, вычислением максимума или минимума).

### **7.2.3 Примерный перечень заданий для решения прикладных задач**

1. Разработайте и реализуйте графический интерфейс пользователя для системы обработки информации по оказанию услуг ЖКХ населению.
2. Реализуйте модель обработки информации по списанию основных средств.
3. Постройте объектную модель формирование цены продажи продукции.
4. Напишите прототип программы учет выплаты стипендии студентам.
5. Постройте UML-диаграммы для программной реализации Обработка информации по расчетам с поставщиками.
6. Разработайте UML-диаграммы для учета вкладчиков в банке.
7. Смоделируйте систему начисления оплаты за дни временной нетрудоспособности.
8. Разработайте программу расчет рейтинга студентов с гибкими правилами
9. Напишите программную реализацию настольной игры
10. Реализуйте предложенную модель программной системы.

### **7.2.4 Примерный перечень вопросов для подготовки к зачету**

Не предусмотрено учебным планом

### **7.2.5 Примерный перечень заданий для подготовки к экзамену**

1. Определите основные принципы объектно-ориентированного программирования.
2. Что такое класс и объект? В чем их взаимосвязь?
3. Объясните понятие инкапсуляции и ее преимущества.
4. Что такое полиморфизм и как он реализуется?
5. Какие преимущества дает наследование?

6. Назовите ключевые слова для объявления класса и объекта на языке Java.

7. Опишите модификаторы доступа и их влияние на видимость членов класса.

8. Проектирование класса. Связанность и зацепление.

9. Интерфейсы класса. Объявление, реализация, применение.

10. Взаимодействие объектов: композиция, агрегация, ассоциация

11. Опишите принципы проектирования SOLID.

12. Что такое шаблон проектирования и приведите примеры.

13. Объясните концепцию связи между объектами.

14. Как организовать иерархию классов для эффективной реализации?

15. Что такое фабричный метод и в каких случаях он применяется?

16. Опишите уровни наследования и их влияние на поведение подклассов.

17. Объясните разницу между полиморфизмом во время компиляции и во время исполнения.

18. Приведите пример переопределения метода и покажите, как он работает.

19. Опишите принцип подстановки Лискова.

20. Построение диаграммы классов на языке UML.

21. Объясните, что такое исключение и как оно обрабатывается.

22. Опишите различные иерархии исключений и их роль.

23. Приведите пример блоков `try-catch` и покажите, как они используются для обработки исключений.

24. Каковы преимущества использования исключений для обработки ошибок?

25. Объясните концепцию иерархии исключений.

26. Опишите, как реализовать интерфейсы для расширения функциональности классов.

27. Объясните концепцию абстрактных классов и их роль в разработке программного обеспечения.

### **7.2.6. Методика выставления оценки при проведении промежуточной аттестации**

Оценка формируется по следующему правилу:

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 6 до 10 баллов

3. Оценка «Хорошо» ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» ставится, если студент набрал от 16 до 20 баллов.)

### 7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Основы объектно-ориентированной парадигмы	ОПК-2, ОПК-4, ОПК-6, ОПК-7	Тест, защита лабораторных работ, выполнение практических заданий и этапов курсового проектирования
2	Абстракция и инкапсуляция	ОПК-2, ОПК-4, ОПК-6, ОПК-7	Тест, защита лабораторных работ, выполнение практических заданий и этапов курсового проектирования
3	Наследование	ОПК-2, ОПК-4, ОПК-6, ОПК-7	Тест, защита лабораторных работ, выполнение практических заданий и этапов курсового проектирования
4	Принципы объектно-ориентированного проектирования	ОПК-2, ОПК-4, ОПК-6, ОПК-7	Тест, защита лабораторных работ, выполнение практических заданий и этапов курсового проектирования
5	Методы и средства программирования	ОПК-2, ОПК-4, ОПК-6, ОПК-7	Тест, защита лабораторных работ, выполнение практических заданий и этапов курсового проектирования

### 7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных

задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

## **8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)**

### **8.1 Перечень учебной литературы, необходимой для освоения дисциплины**

1. Гуськова, О. И. *Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. — 2-е изд. — Москва : Московский педагогический государственный университет, 2024. — 240 с. — ISBN 978-5-4263-0648-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/146115.html> (дата обращения: 01.12.2024)*

2. Мейер, Б. *Объектно-ориентированное программирование и программная инженерия : учебное пособие / Б. Мейер. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 284 с. — ISBN 978-5-4497-2464-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/133956.html> (дата обращения: 21.11.2023).*

3. Ермаков, А. В. *Объектно-ориентированное программирование в задачах на языке Java : учебное пособие / А. В. Ермаков. — Саратов : Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2022. — 156 с. — ISBN 978-5-7433-3478-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/128034.html> (дата обращения: 31.01.2025).*

**8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:**

*Программное обеспечение:*

*Среда разработки ПО (IDE IntelliJ IDEA)*

*Информационно-справочные системы:*

1. Сайт CITForum [www.citforum.ru](http://www.citforum.ru) Библиотека технических материалов по информационным технологиям

2. Сайт компании Sun для поддержки языка программирования Java <http://docs.oracle.com/javase/8/javafx/get-started-tutorial/> - описание технологии JavaFx

3. Сайт поддержки разработчиков ПО <https://www.jetbrains.com/ru-ru/support/>, Справочная техническая документация по среде разработки IntelliJ IDEA и поддержки языка программирования Java

4. Комитет по стандартизации в области радиоэлектроники и вычислительной техники [www.ieee.org](http://www.ieee.org) Нормативно-справочная документация по информатике, программированию и вычислительной техники

5. SWEBOK (Software Engineering Body of Knowledge) <http://www.swebok.org/> международный стандарт ISO/IEC TR 19759, в котором описана общепринятая сумма знаний по программной инженерии <https://ligurio.github.io/swebok-ru/> — в авторской интерпретации (С. Орлик 2014 г.)

## **9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

*Компьютерный класс с предустановленной средой разработки приложений.*

*Мультимедийные средства: наборы файлов презентаций по темам лекционных занятий, комплект видеороликов по установке, настройке и примерам использования инструментальных средств технологии программирования.*

*Средства мониторинга – программа тестирования по модулям дисциплины с базами тестовых вопросов*

## **10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

По дисциплине «Объектно-ориентированное программирование» читаются лекции, проводятся практические занятия и лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Практические занятия направлены на приобретение практических навыков выполнения работ по созданию программного продукта с использованием инструментальных средств. Занятия проводятся путем решения конкретных задач в аудитории.

Лабораторные работы выполняются на специально ПО в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится защитой курсового проекта.

Вид учебных занятий	Деятельность студента
---------------------	-----------------------

Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Практическое занятие	Конспектирование рекомендуемых источников. Работа с конспектом лекций, подготовка ответов к контрольным вопросам, просмотр рекомендуемой литературы. Прослушивание аудио- и видеозаписей по заданной теме, выполнение расчетно-графических заданий, решение задач по алгоритму.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начинаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом с оценкой, зачетом с оценкой, экзаменом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	----------------------------	--