

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Агафонов Александр Викторович

Должность: директор филиала

Дата подписания: 19.06.2025 15:33:26

Уникальный программный ключ:

23E0K5AR5K5I5N5T5U5T405

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**  
**ВЫСШЕГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**ЧЕБОКСАРСКИЙ ИНСТИТУТ (ФИЛИАЛ) МОСКОВСКОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА**

**Кафедра Информационных технологий**  
**и систем управления**



## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

### **«Объектно-ориентированное программирование»**

(наименование дисциплины)

Направление подготовки	<b>09.03.01 «Информатика и вычислительная техника»</b> (код и наименование направления подготовки)
Направленность (профиль) подготовки	<b>«Программное обеспечение вычислительной техники и автоматизированных систем»</b> (наименование профиля подготовки)
Квалификация выпускника	<b>бакалавр</b>
Форма обучения	<b>очная, заочная</b>
Год начала обучения	<b>2025</b>

Рабочая программа дисциплины разработана в соответствии с:

- Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 – Информатика и вычислительная техника, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 929 от 19 сентября 2017 г. зарегистрированный в Минюсте 10 октября 2017 года, рег. номер 48489 (далее – ФГОС ВО).

- учебным планом (очной, заочной форм обучения) по направлению подготовки 09.03.01 «Информатика и вычислительная техника».

Рабочая программ дисциплины включает в себя оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине (п.8 Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины)

Автор Пикина Наталия Евгеньевна, кандидат педагогических наук, доцент кафедры информационных технологий и систем управления

*(указать ФИО, ученую степень, ученое звание или должность)*

Программа одобрена на заседании кафедры Информационных технологий и систем управления (протокол № 8 от 12.04.2025г).

**1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы (Цели освоения дисциплины)**

1.1. Целями освоения дисциплины «Объектно-ориентированное программирование» являются:

- ознакомление студентов с организацией современных операционных систем, с организацией управления и взаимодействия процессов, организацией управления в многопользовательских и многозадачных системах. Также рассматриваются архитектуры современных параллельных вычислительных систем, дается обзор современных операционных систем.

1.2. Области профессиональной деятельности и(или) сферы профессиональной деятельности, в которых выпускники, освоившие программу, могут осуществлять профессиональную деятельность:

*Об Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации средств вычислительной техники и информационных систем, управления их жизненным циклом).*

1.3. К основным задачам изучения дисциплины относится подготовка обучающихся к выполнению трудовых функций в соответствии с профессиональными стандартами:

Код и наименование профессионального стандарта	Обобщенные трудовые функции			Трудовые функции		
	код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
06.001 Программист Профессиональный стандарт «Программист», утвержденный приказом Министерством труда и социальной защиты Российской Федерации от 18.11.2013 № 679н (зарегистрирован в Министерстве труда и социальной защиты Российской Федерации от 18 декабря 2013 г. №30635)	D	Разработка требований и проектирование программного обеспечения	6	Анализ требований к программному обеспечению	D/01.6	6
			6	Разработка технических спецификаций на программные компоненты и их взаимодействие	D/02.6	
			6	Проектирование программного обеспечения	D/03.6	
06.028 Системный программист Профессиональный стандарт «Системный программист», утв. Приказом	A	Разработка компонентов системных программных продуктов	6	Разработка драйверов устройств	A/01.6	6
			6	Разработка компиляторов, загрузчиков,	A/02.6	6

Код и наименование профессионального стандарта	Обобщенные трудовые функции			Трудовые функции		
	код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
Министерством труда и социальной защиты РФ от 29 сентября 2020 года N 678н				сборщиков		
				Разработка системных утилит	A/03.6	6
				Создание инструментальных средств программирования	A/04.6	6
06.015 Специалист по информационным системам Профессиональный стандарт «Специалист по информационным системам», утв. приказом Министерством труда и социальной защиты РФ 18 ноября 2014 г. №896н	С	Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	6	Разработка модели бизнес-процессов заказчика	C/08.6	6
				Выявление требований к ИС	C/11.6	6
				Анализ требований	C/12.6	6
				Разработка архитектуры ИС	C/14.6	6
				Проектирование и дизайн ИС	C/16.6	6
				Разработка баз данных ИС	C/17.6	6
				Организационное и технологическое обеспечение кодирования на языках программирования	C/18.6	6
				Организационное и технологическое обеспечение модульного тестирования ИС (верификации)	C/19.6	6
				Создание пользовательской документации к ИС	C/22.6	6
				Организация репозитория хранения данных о создании (модификации) и вводе ИС в эксплуатацию	C/40.6	6

1.4. Компетенции обучающегося, формируемые в результате освоения дисциплины

Наименование категории (группы) компетенций	Код и наименование компетенций	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения
Разработка алгоритмов	ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1. Знать: основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения.	<p><i>на уровне знаний:</i> основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения.</p> <p><i>на уровне умений:</i> уметь разрабатывать алгоритмы.</p> <p><i>на уровне навыков:</i> владение современными программными средствами для решения практических задач в своей профессиональной деятельности</p>
		ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули.	<p><i>на уровне знаний:</i> знать основные концепции объектно-ориентированного программирования.</p> <p><i>на уровне умений:</i> составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули.</p> <p><i>на уровне навыков:</i> владение навыками писать и отлаживать коды на языке программирования, тестировать работоспособность программы.</p>
		ОПК-8.3. Владеть: языком программирования, методами отладки и тестирования работоспособности программы	<p><i>на уровне знаний:</i> знать языки и технологии программирования, пригодные для практического применения в области медиаиндустрии и дизайна</p> <p><i>на уровне умений:</i> уметь использовать методы отладки и тестирования работоспособности программы.</p>

			<p><i>на уровне навыков:</i>  владение языком программирования, методами отладки и тестирования работоспособности программы.</p>
<p>Разработка, отладка, модификация и поддержка системного программного обеспечения</p>	<p>ПК-3. Способен разрабатывать компоненты системных программных продуктов</p>	<p>ПК-3.1 Разрабатывает драйверы устройств</p>	<p><i>на уровне знаний:</i>  знать общие принципы работы программных средств под управлением современных операционных систем.  <i>на уровне умений:</i>  уметь разрабатывать объектно-ориентированную модель предметной области.  <i>на уровне навыков:</i>  владение современными программными средствами для решения практических задач в своей профессиональной деятельности.</p>
		<p>ПК-3.2 Разрабатывает компиляторы, загрузчики, сборщики</p>	<p><i>на уровне знаний:</i>  знать основные концепции объектно-ориентированного программирования.  <i>на уровне умений:</i>  уметь осваивать и применять программные средства для решения практических задач в своей профессиональной деятельности.  <i>на уровне навыков:</i>  владение навыками самостоятельного овладения новыми технологиями, новыми знаниями по платформам объектно-ориентированных CASE</p>

		ПК-3.3 Разрабатывает системные утилиты	<i>на уровне знаний:</i> знать языки и технологии программирования, пригодные для практического применения. <i>на уровне умений:</i> уметь использовать современные информационные технологии. <i>на уровне навыков:</i> владение навыками применения современных программных средств.
--	--	----------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2. Место дисциплины в структуре ОПОП

Дисциплина Б.Д(М).Б.26 «Объектно-ориентированное программирование» реализуется в рамках обязательной части Блока 1 «Дисциплины (модули)» программы бакалавриата.

Дисциплина «Объектно-ориентированное программирование» преподается обучающимся по очной форме обучения – в 4-м семестре, по заочной форме – в 9-м семестре.

Дисциплина «Объектно-ориентированное программирование» является промежуточным этапом формирования компетенций ОПК-8, ПК-3 в процессе освоения ОПОП.

Дисциплина «Объектно-ориентированное программирование» основывается на знаниях, умениях и навыках, приобретенных при изучении дисциплин Перспективы развития информатики и вычислительной техники, Введение в информатику, Учебная практика: ознакомительная практика и является предшествующей для изучения дисциплин Программирование для мобильных устройств, Структуры и алгоритмы обработки данных, Криптографические методы защиты информации, Защита информации, Учебная практика: научно-исследовательская работа (получение первичных навыков научно-исследовательской работы), Учебная практика: технологическая (проектно-технологическая) практика, Микропроцессорные устройства систем управления, Интернет-технологии, Интернет-программирование, Функциональное и логическое программирование, Системное программирование, Государственная итоговая аттестация: подготовка к сдаче и сдача государственного экзамена, Государственная итоговая аттестация: выполнение и защита выпускной квалификационной работы.

Формой промежуточной аттестации знаний обучаемых по очной форме обучения является экзамен в 4-м семестре, по заочной форме экзамен в 9-м семестре.

## 3. Объем дисциплины

очная форма обучения:

Вид учебной работы по дисциплине	Всего в з.е. и часах	Семестр 4 в часах
----------------------------------	----------------------	----------------------

<b>Общая трудоёмкость дисциплины</b>	<b>5 з.е. -180 ак.час</b>	<b>180 ак.час</b>
<b>Контактная работа - Аудиторные занятия</b>	<b>73</b>	<b>73</b>
<i>Лекции</i>	36	36
<i>Лабораторные занятия</i>	36	36
<i>Семинары, практические занятия</i>	-	-
<i>Консультация</i>	1	1
<b>Самостоятельная работа</b>	<b>71</b>	<b>71</b>
<b>Курсовая работа (курсовой проект)</b>	-	-
<b>Вид промежуточной аттестации</b>	Экзамен – 36 часов	Экзамен – 36 часов

заочная форма обучения:

<b>Вид учебной работы по дисциплине</b>	<b>Всего в з.е. и часах</b>	<b>Семестр 9 в часах</b>
<b>Общая трудоёмкость дисциплины</b>	<b>5 з.е. -180 ак.час</b>	<b>180 ак.час</b>
<b>Контактная работа - Аудиторные занятия</b>	<b>13</b>	<b>13</b>
<i>Лекции</i>	6	6
<i>Лабораторные занятия</i>	6	6
<i>Семинары, практические занятия</i>	-	-
<i>Консультация</i>	1	1
<b>Самостоятельная работа</b>	<b>158</b>	<b>158</b>
<b>Курсовая работа (курсовой проект)</b>	-	-
<b>Вид промежуточной аттестации</b>	Экзамен – 9 часов	Экзамен – 9 часов

**4. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий**

**4.1. Учебно-тематический план**

Очная форма обучения

Тема (раздел)	Количество часов				Код индикатора достижений компетенции
	контактная работа			самостоятельная работа	
	лекции и	лабораторные занятия	семинары и практические занятия		
Тема 1. Введение в объектно-ориентированное программирование. Введение в объектно-ориентированное проектирование.	8	8	-	17	ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3
Тема 2. Классы и объекты. Работа с объектами	8	8	-	18	ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3
Тема 3. Наследование. Одиночное и множественное наследование. Полиморфизм. Виртуальные методы	10	10	-	18	ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3

Тема 4. Обработка исключений. Шаблоны классов. Поточковые классы. Контейнерные классы.	10	10	-	18	ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3
Консультация	1			-	ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3
Контроль (экзамен)	36				ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3
<b>ИТОГО</b>	<b>73</b>			<b>71</b>	

### Заочная форма обучения

Тема (раздел)	Количество часов				Код индикатора достижений компетенции
	контактная работа			самостоятельная работа	
	лекции	лабораторные занятия	семинары и практические занятия		
Тема 1. Введение в объектно-ориентированное программирование. Введение в объектно-ориентированное проектирование.	2	-	-	38	ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3
Тема 2. Классы и объекты. Работа с объектами	2	2	-	40	ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3
Тема 3. Наследование. Одиночное и множественное наследование. Полиморфизм. Виртуальные методы	2	2	-	40	ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3
Тема 4. Обработка исключений. Шаблоны классов. Поточковые классы. Контейнерные классы.	-	2	-	40	ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3
Консультация	1			-	ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3
Контроль (экзамен)	9				ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК-3.1, ПК-3.2, ПК-3.3

<b>ИТОГО</b>	<b>13</b>	<b>158</b>	
--------------	-----------	------------	--

## **4.2. Содержание дисциплины**

### **Тема 1. Введение в объектно-ориентированное программирование.**

#### **Введение в объектно-ориентированное проектирование.**

История развития и основные парадигмы программирования. Отличия ООП от процедурного подхода.

Принципы ООП: инкапсуляция, наследование, полиморфизм, абстракция.

Цели и преимущества объектно-ориентированного подхода в разработке программного обеспечения.

Введение в UML-диаграммы и объектно-ориентированное проектирование.

Понятие объекта и класса как ключевых понятий ООП. Связь между реальными объектами и программными структурами.

### **Тема 2. Классы и объекты. Работа с объектами**

Объявление и определение классов. Конструкторы и деструкторы.

Атрибуты (поля) и методы классов. Модификаторы доступа: public, private, protected.

Создание объектов и обращение к их членам. Использование указателей и ссылок на объекты.

Перегрузка методов и операторов. Основы инкапсуляции и сокрытия реализации.

Статические поля и методы. Использование ключевого слова this.

### **Тема 3. Наследование. Одиночное и множественное наследование.**

#### **Полиморфизм. Виртуальные методы**

Понятие наследования. Базовые и производные классы.

Механизм одиночного и множественного наследования. Проблемы и способы их решения (алмазное наследование).

Полиморфизм: статический (перегрузка) и динамический (виртуальные функции).

Виртуальные методы, таблицы виртуальных функций (vtable), переопределение методов.

Абстрактные классы и интерфейсы. Применение в проектировании и архитектуре программ.

Примеры использования наследования и полиморфизма в реальных проектах.

### **Тема 4. Обработка исключений. Шаблоны классов. Поточные классы. Контейнерные классы.**

Обработка ошибок в ООП. Механизмы исключений: try, catch, throw, finally.

Пользовательские типы исключений. Преимущества централизованной обработки ошибок.

Введение в шаблоны (templates): шаблоны функций и классов.

Использование шаблонов для создания универсальных структур и алгоритмов.

Потоковые классы для ввода/вывода (iostream, fstream). Работа с файлами.

Контейнерные классы стандартной библиотеки (vector, list, map, set).

Итераторы и алгоритмы.

Принципы обобщённого программирования и их применение в ООП.

## **5. Учебно-методическое обеспечение самостоятельной работы студентов**

Самостоятельная работа проводится с целью: систематизации и закрепления полученных теоретических знаний и практических умений обучающихся; углубления и расширения теоретических знаний студентов; формирования умений использовать нормативную, правовую, справочную документацию, учебную и специальную литературу; развития познавательных способностей и активности обучающихся: творческой инициативы, самостоятельности, ответственности, организованности; формирование самостоятельности мышления, способностей к саморазвитию, совершенствованию и самоорганизации; формирования профессиональных компетенций; развитию исследовательских умений студентов.

Формы и виды самостоятельной работы студентов: чтение основной и дополнительной литературы – самостоятельное изучение материала по рекомендуемым литературным источникам; работа с библиотечным каталогом, самостоятельный подбор необходимой литературы; работа со словарем, справочником; поиск необходимой информации в сети Интернет; конспектирование источников; реферирование источников; составление аннотаций к прочитанным литературным источникам; составление рецензий и отзывов на прочитанный материал; составление обзора публикаций по теме; составление и разработка терминологического словаря; составление хронологической таблицы; составление библиографии (библиографической картотеки); подготовка к различным формам текущей и промежуточной аттестации (к тестированию, контрольной работе, экзамену); выполнение домашних контрольных работ; самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, задачи, тесты; выполнение творческих заданий).

Технология организации самостоятельной работы обучающихся включает использование информационных и материально-технических ресурсов образовательного учреждения: библиотеку с читальным залом, компьютерные классы с возможностью работы в Интернет; аудитории (классы) для консультационной деятельности.

Перед выполнением обучающимися внеаудиторной самостоятельной работы преподаватель проводит консультирование по выполнению задания, который включает цель задания, его содержания, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. Во время выполнения обучающимися внеаудиторной

самостоятельной работы и при необходимости преподаватель может проводить индивидуальные и групповые консультации.

Контроль самостоятельной работы студентов предусматривает: соотнесение содержания контроля с целями обучения; объективность контроля; валидность контроля (соответствие предъявляемых заданий тому, что предполагается проверить); дифференциацию контрольно-измерительных материалов.

Формы контроля самостоятельной работы: просмотр и проверка выполнения самостоятельной работы преподавателем; организация самопроверки, взаимопроверки выполненного задания в группе; обсуждение результатов выполненной работы на занятии; проведение письменного опроса; проведение устного опроса; организация и проведение индивидуального собеседования; организация и проведение собеседования с группой.

### **Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы**

<b>Наименование тем (разделов) дисциплины</b>	<b>Перечень вопросов, отводимых на самостоятельное освоение</b>	<b>Формы внеаудиторной самостоятельной работы</b>
Тема 1. Введение в объектно-ориентированное программирование . Введение в объектно-ориентированное проектирование.	<ol style="list-style-type: none"> <li>1. Основные концепции объектно-ориентированного программирования (ООП).</li> <li>2. Преимущества и недостатки ООП по сравнению с процедурным программированием.</li> <li>3. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.</li> <li>4. Объектно-ориентированное проектирование: принципы и методы.</li> <li>5. Роль UML в объектно-ориентированном проектировании.</li> <li>6. Основные подходы к проектированию классов и объектов.</li> <li>7. Паттерны проектирования: определение и примеры использования.</li> <li>8. Применение ООП в современных языках программирования.</li> <li>9. Этапы разработки программного обеспечения с использованием ООП.</li> <li>10. Будущее объектно-ориентированного программирования: новые подходы и технологии.</li> </ol>	Изучение литературы по основам ООП и его принципам. Проведение сравнительного анализа различных подходов к объектно-ориентированному проектированию.
Тема 2. Классы и объекты. Работа с объектами	<ol style="list-style-type: none"> <li>1. Определение классов и объектов в контексте ООП.</li> <li>2. Синтаксис определения классов и создания объектов в различных языках программирования.</li> <li>3. Конструкторы и деструкторы: их роль и применение.</li> <li>4. Атрибуты и методы классов: объявление и использование.</li> <li>5. Инкапсуляция данных: модификаторы доступа и их значение.</li> </ol>	Решение практических задач на создание и использование классов. Проведение анализа инкапсуляции в примерах программ.

	<p>6. Практические примеры работы с классами и объектами.</p> <p>7. Сравнение статических и динамических методов классов.</p> <p>8. Взаимодействие объектов: сообщения и вызовы методов.</p> <p>9. Применение классов в реальных приложениях: примеры.</p> <p>10. Будущее классов и объектов в программировании: новые тренды.</p>	
<p>Тема 3. Наследование. Одиночное и множественное наследование. Полиморфизм. Виртуальные методы</p>	<p>1. Определение наследования и его роль в ООП.</p> <p>2. Одиночное и множественное наследование: преимущества и недостатки.</p> <p>3. Полиморфизм: определение и примеры использования.</p> <p>4. Виртуальные методы и их применение в наследовании.</p> <p>5. Переопределение методов: как это работает и когда используется.</p> <p>6. Примеры использования наследования в реальных проектах.</p> <p>7. Проблемы, возникающие при множественном наследовании.</p> <p>8. Интерфейсы и абстрактные классы: как они связаны с наследованием.</p> <p>9. Практические примеры полиморфизма в различных языках программирования.</p> <p>10. Будущее наследования и полиморфизма: новые подходы и технологии.</p>	<p>Исследование примеров использования виртуальных методов и их значимости. Решение задач на переопределение методов в наследуемых классах.</p>
<p>Тема 4. Обработка исключений. Шаблоны классов. Потоковые классы. Контейнерные классы.</p>	<p>1. Определение исключений и их роль в программировании.</p> <p>2. Механизмы обработки исключений: try, catch, finally.</p> <p>3. Создание пользовательских исключений в различных языках программирования.</p> <p>4. Шаблоны классов: определение и применение.</p> <p>5. Примеры использования шаблонов классов в реальных проектах.</p> <p>6. Потоковые классы: работа с потоками ввода-вывода.</p> <p>7. Контейнерные классы: определение и основные типы (списки, множества, словари).</p> <p>8. Применение контейнерных классов для хранения и обработки данных.</p> <p>9. Взаимосвязь между потоковыми и контейнерными классами.</p> <p>10. Будущее обработки исключений и шаблонов классов: новые подходы и технологии.</p>	<p>Решение задач на создание пользовательских исключений и их обработку. Проведение анализа примеров использования шаблонов классов.</p>

### Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему самостоятельной работы, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер
«Хорошо»	Обучающийся в целом раскрывает тему

	самостоятельной работы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера
«Удовлетворительно»	Обучающийся в целом раскрывает тему самостоятельной работы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой самостоятельной работы

**6. Оценочные материалы (фонд оценочных средств) для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины**

**6.1. Паспорт фонда оценочных средств**

№	Контролируемые разделы (темы) дисциплины	Код и наименование компетенции	Индикатор достижения компетенции	Наименование оценочного средства
1.	Тема 1. Введение в объектно-ориентированное программирование. Введение в объектно-ориентированное проектирование.	ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1. Знать: основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения. ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули. ОПК-8.3. Владеть: языком программирования, методами отладки и тестирования работоспособности программы	Опрос, доклад, тест реферат, зачет
		ПК-3. Способен разрабатывать компоненты системных программных продуктов	ПК-3.1 Разрабатывает драйверы устройств ПК-3.2 Разрабатывает компиляторы, загрузчики, сборщики ПК-3.3 Разрабатывает	

			системные утилиты ПК-3.4. Создает инструментальные средства программирования	
2.	Тема 2. Классы и объекты. Работа с объектами	ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1. Знать: основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения. ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули. ОПК-8.3. Владеть: языком программирования, методами отладки и тестирования работоспособности программы	Опрос, доклад, тест реферат, зачет
		ПК-3. Способен разрабатывать компоненты системных программных продуктов	ПК-3.1 Разрабатывает драйверы устройств ПК-3.2 Разрабатывает компиляторы, загрузчики, сборщики ПК-3.3 Разрабатывает системные утилиты ПК-3.4. Создает инструментальные средства программирования	
3.	Тема 3. Наследование. Одиночное и множественное наследование. Полиморфизм. Виртуальные методы	ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1. Знать: основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения. ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке	Опрос, доклад, тест реферат, зачет

			<p>программирования, тестировать работоспособность программы, интегрировать программные модули. ОПК-8.3. Владеть: языком программирования, методами отладки и тестирования работоспособности программы</p>	
		ПК-3. Способен разрабатывать компоненты системных программных продуктов	<p>ПК-3.1 Разрабатывает драйверы устройств  ПК-3.2 Разрабатывает компиляторы, загрузчики, сборщики  ПК-3.3 Разрабатывает системные утилиты  ПК-3.4. Создает инструментальные средства программирования</p>	
4.	Тема 4. Обработка исключений. Шаблоны классов. Поточные классы. Контейнерные классы.	ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	<p>ОПК-8.1. Знать: основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения.  ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули.  ОПК-8.3. Владеть: языком программирования, методами отладки и тестирования работоспособности программы</p>	Опрос, доклад, тест реферат, зачет
		ПК-3. Способен разрабатывать компоненты системных программных	<p>ПК-3.1 Разрабатывает драйверы устройств  ПК-3.2 Разрабатывает компиляторы,</p>	

		продуктов	загрузчики, сборщики ПК-3.3 Разрабатывает системные утилиты ПК-3.4. Создает инструментальные средства программирования	
--	--	-----------	---------------------------------------------------------------------------------------------------------------------------------------------	--

**Этапы формирования компетенций в процессе освоения ОПОП** прямо связаны с местом дисциплин в образовательной программе. Каждый этап формирования компетенции, характеризуется определенными знаниями, умениями и навыками и (или) опытом профессиональной деятельности, которые оцениваются в процессе текущего контроля успеваемости, промежуточной аттестации по дисциплине (практике) и в процессе итоговой аттестации.

Дисциплина «Объектно-ориентированное программирование» является промежуточным этапом комплекса дисциплин, в ходе изучения которых у студентов формируются компетенции ОПК-8, ПК-3.

Формирование компетенции ОПК-8, ПК-3 начинается с освоения дисциплин «Перспективы развития информатики и вычислительной техники», «Введение в информатику», Учебная практика (ознакомительная практика).

Завершается работа по формированию у студентов указанных компетенций в ходе изучения дисциплин «Программирование для мобильных устройств», «Структуры и алгоритмы обработки данных», «Криптографические методы защиты информации», «Защита информации», Учебная практика: научно-исследовательская работа (получение первичных навыков научно-исследовательской работы), Учебная практика: технологическая (проектно-технологическая) практика, «Микропроцессорные устройства систем управления», «Интернет-технологии», «Интернет-программирование», «Функциональное и логическое программирование», «Системное программирование».

Итоговая оценка сформированности компетенций ОПК-8, ПК-3 определяется в период Государственная итоговая аттестация: подготовка к сдаче и сдача государственного экзамена, Государственная итоговая аттестация: выполнение и защита выпускной квалификационной работы.

**В процессе изучения дисциплины, компетенции также формируются поэтапно.**

Основными этапами формирования ОПК-8, ПК-3 при изучении дисциплины Б1.Д(М).Б.26 «Объектно-ориентированное программирование» является последовательное изучение содержательно связанных между собой тем учебных занятий. Изучение каждой темы предполагает овладение студентами необходимыми дескрипторами (составляющими) компетенций. Для оценки уровня сформированности компетенций в процессе изучения дисциплины предусмотрено проведение текущего контроля успеваемости по

темам (разделам) дисциплины и промежуточной аттестации по дисциплине – экзамен.

## 6.2. Контрольные задания и материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

### 6.2.1. Контрольные вопросы по темам (разделам) для опроса на занятиях

Тема (раздел)	Вопросы
Тема 1. Введение в объектно-ориентированное программирование. Введение в объектно-ориентированное проектирование.	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Основные принципы объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм).</li> <li>2. Отличие процедурного программирования от объектно-ориентированного.</li> <li>3. Понятие объекта и класса в контексте ООП.</li> <li>4. Роль абстракции в объектно-ориентированном подходе.</li> <li>5. Этапы объектно-ориентированного проектирования.</li> </ol> <p>ПК-3</p> <ol style="list-style-type: none"> <li>6. Диаграммы классов и их использование при проектировании программ.</li> <li>7. Концепция модульности в объектно-ориентированной архитектуре.</li> <li>8. Объектно-ориентированная модель данных и ее особенности.</li> <li>9. Преимущества повторного использования кода в ООП.</li> <li>10. Связь между ООП и методологиями разработки ПО (например, UML, Agile).</li> </ol>
Тема 2. Классы и объекты. Работа с объектами	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Объявление и реализация класса в ООП-языках.</li> <li>2. Создание и инициализация объектов на основе класса.</li> <li>3. Использование конструкторов и деструкторов в классах.</li> <li>4. Разграничение доступа с помощью модификаторов public, private, protected.</li> <li>5. Работа с методами и свойствами объектов.</li> </ol> <p>ПК-3</p> <ol style="list-style-type: none"> <li>6. Использование указателей и ссылок на объекты.</li> <li>7. Перегрузка методов и операторов.</li> <li>8. Использование статических членов класса.</li> <li>9. Размещение объектов в динамической и статической памяти.</li> <li>10. Примеры организации взаимодействия между объектами.</li> </ol>
Тема 3. Наследование. Одиночное и множественное наследование. Полиморфизм. Виртуальные методы	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Понятие наследования и его роль в повторном использовании кода.</li> <li>2. Реализация одиночного и множественного наследования.</li> <li>3. Разрешение конфликтов при множественном наследовании.</li> <li>4. Использование ключевых слов virtual и override.</li> <li>5. Механизм полиморфизма и его реализация в ООП-языках.</li> </ol>

	<p>ПК-3</p> <ol style="list-style-type: none"> <li>6. Виртуальные функции и таблицы виртуальных методов (vtable).</li> <li>7. Различие между статическим и динамическим полиморфизмом.</li> <li>8. Примеры иерархий классов с использованием наследования.</li> <li>9. Абстрактные классы и чисто виртуальные функции.</li> <li>10. Роль наследования и полиморфизма в проектировании гибких систем.</li> </ol>
<p>Тема 4. Обработка исключений. Шаблоны классов. Поточные классы. Контейнерные классы.</p>	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Основы обработки исключений с использованием конструкции try-catch.</li> <li>2. Создание и выбрасывание собственных исключений.</li> <li>3. Иерархия классов исключений в стандартной библиотеке.</li> <li>4. Понятие шаблонов классов и их синтаксис.</li> <li>5. Использование шаблонов для реализации обобщённых структур данных.</li> </ol> <p>ПК-3</p> <ol style="list-style-type: none"> <li>6. Работа с потоками ввода-вывода (I/O streams) в стандартной библиотеке.</li> <li>7. Классы ifstream, ofstream и fstream для работы с файлами.</li> <li>8. Назначение и примеры использования контейнерных классов (vector, list, map и др.).</li> <li>9. Итераторы и алгоритмы при работе с контейнерами.</li> <li>10. Практические примеры сочетания шаблонов, исключений и контейнеров в одном проекте.</li> </ol>

### Шкала оценивания ответов на вопросы

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает ответ на каждый теоретический вопрос, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает теоретические вопросы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает теоретические вопросы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не знает ответов на поставленные теоретические вопросы.

### 6.2.2. Темы для докладов

Тема (раздел)	Вопросы
<p>Тема 1. Введение в объектно-ориентированное программирование. Введение в объектно-ориентированное проектирование.</p>	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Основные принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм.</li> <li>2. История и эволюция объектно-ориентированных языков программирования.</li> <li>3. Объектно-ориентированное проектирование: от теории к практике.</li> <li>4. Преимущества и недостатки объектно-ориентированного подхода</li> </ol>

	<p>в программировании. ПК-3</p> <ol style="list-style-type: none"> <li>5. Роль UML (Unified Modeling Language) в объектно-ориентированном проектировании.</li> <li>6. Сравнение объектно-ориентированного и процедурного подходов в программировании.</li> <li>7. Принципы SOLID и их применение в объектно-ориентированном проектировании.</li> <li>8. Разработка паттернов проектирования для эффективного объектно-ориентированного кода.</li> </ol>
<p>Тема 2. Классы и объекты. Работа с объектами</p>	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Что такое класс и объект? Различия и связь между ними.</li> <li>2. Конструкторы и деструкторы классов в объектно-ориентированном программировании.</li> <li>3. Принципы инкапсуляции: скрытие данных и методы доступа.</li> <li>4. Работа с объектами в C++, Java и Python: особенности.</li> </ol> <p>ПК-3</p> <ol style="list-style-type: none"> <li>5. Управление памятью при работе с объектами: создание и удаление.</li> <li>6. Проблемы утечек памяти при работе с объектами и способы их предотвращения.</li> <li>7. Статические и нестатические члены класса: их различия и применение.</li> <li>8. Особенности работы с объектами в многозадачных и многопоточных приложениях.</li> </ol>
<p>Тема 3. Наследование. Одиночное и множественное наследование. Полиморфизм. Виртуальные методы</p>	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Понятие наследования в объектно-ориентированном программировании.</li> <li>2. Различие между одиночным и множественным наследованием.</li> <li>3. Проблемы множественного наследования и их решение (например, через интерфейсы).</li> <li>4. Полиморфизм в объектно-ориентированном программировании: понятие и примеры.</li> </ol> <p>ПК-3</p> <ol style="list-style-type: none"> <li>5. Виртуальные и чисто виртуальные методы: что это и как их правильно использовать.</li> <li>6. Абстракция и интерфейсы в объектно-ориентированных языках программирования.</li> <li>7. Применение полиморфизма в реальных приложениях (пример с коллекциями).</li> <li>8. Принципы и преимущества использования паттернов проектирования, основанных на наследовании.</li> </ol>
<p>Тема 4. Обработка исключений. Шаблоны классов. Поточные классы. Контейнерные классы.</p>	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Основы обработки исключений в объектно-ориентированных языках программирования.</li> <li>2. Обработка исключений в C++ и Java: синтаксис и основные принципы.</li> <li>3. Как правильно использовать try-catch блоки для обработки ошибок.</li> <li>4. Шаблоны классов: что это такое и где они применяются.</li> </ol> <p>ПК-3</p> <ol style="list-style-type: none"> <li>5. Мощь шаблонов классов: примеры использования в реальных</li> </ol>

	<p>проектах.</p> <p>6. Поточковые классы: как работают потоки и их синхронизация в объектно-ориентированном программировании.</p> <p>7. Контейнерные классы в C++ STL и Java Collections Framework.</p> <p>8. Паттерны проектирования для обработки исключений в многозадачных системах.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему доклада, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает тему доклада, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает тему доклада и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой

### 6.2.3. Оценочные средства остаточных знаний (тест)

#### ОПК-8

**1. Какой принцип ООП описывает скрытие реализации и предоставление только интерфейса?**

- 1) Наследование
- 2) Инкапсуляция
- 3) Полиморфизм
- 4) Абстракция

**2. Что такое полиморфизм в ООП?**

- 1) Принцип скрытия данных
- 2) Способность объектов разных классов обрабатывать одно и то же сообщение
- 3) Создание классов, которые наследуются от других классов
- 4) Принцип, который запрещает изменение состояний объектов

**3. Какой оператор используется для указания, что класс наследует другой класс в языке Java?**

- 1) implements
- 2) extends
- 3) super
- 4) interface

**4. Какая из следующих характеристик относится к инкапсуляции?**

- 1) Доступность данных только внутри класса
- 2) Доступность данных только для наследников
- 3) Принцип, который позволяет создавать новые классы

4) Способность объектов взаимодействовать друг с другом

**5. Как называется метод, который имеет ту же сигнатуру, что и метод родительского класса, но с другим поведением?**

- 1) Перегрузка метода
- 2) Переопределение метода
- 3) Инкапсуляция
- 4) Скрытие данных

**6. Что такое абстракция в ООП?**

- 1) Способность класса скрывать детали реализации и показывать только важные для пользователя элементы
- 2) Способность объектов обрабатывать одно и то же сообщение
- 3) Способность классов изменять свои состояния
- 4) Способность к созданию новых классов

**7. Что происходит, если в классе не реализован конструктор по умолчанию?**

- 1) Он автоматически создается компилятором
- 2) Программа не скомпилируется
- 3) Класс не может быть инстанцирован
- 4) Компилятор выдает ошибку при вызове конструктора

**8. Какой ключевой момент в наследовании позволяет классам наследовать поведение и свойства другого класса?**

- 1) Наследование абстрактных классов
- 2) Использование полиморфизма
- 3) Использование ключевого слова `extends`
- 4) Перегрузка методов

**9. Что делает ключевое слово `super` в Java?**

- 1) Создает новый объект родительского класса
- 2) Позволяет вызвать конструктор родительского класса
- 3) Определяет тип объекта
- 4) Останавливает выполнение программы

**10. Как в Java объявляется абстрактный класс?**

- 1) `class MyClass {}`
- 2) `abstract class MyClass {}`
- 3) `interface MyClass {}`
- 4) `class abstract MyClass {}`

**11. Какой принцип ООП обеспечивает возможность использования объектов разных типов через одинаковый интерфейс?**

- 1) Наследование

- 2) Полиморфизм
- 3) Инкапсуляция
- 4) Абстракция

**12. Что делает перегрузка методов?**

- 1) Создает методы с разными именами
- 2) Создает методы с одинаковыми именами, но разными параметрами
- 3) Создает методы, которые могут быть вызваны только один раз
- 4) Осуществляет полное переопределение методов

**13. Какой из принципов ООП не является обязательным для создания классов в языке Java?**

- 1) Инкапсуляция
- 2) Наследование
- 3) Абстракция
- 4) Перегрузка

**14. Что такое интерфейс в ООП?**

- 1) Класс, который не имеет реализации
- 2) Механизм, позволяющий описать контракт для классов
- 3) Способ реализации полиморфизма
- 4) Структура данных, аналогичная классу

**15. Какое из утверждений верно относительно частных (private) методов и свойств в Java?**

- 1) Они доступны только внутри класса
- 2) Они доступны для всех классов
- 3) Они доступны только в дочерних классах
- 4) Они доступны только для объектов этого класса

**16. Какое из утверждений относительно абстрактных классов верно?**

- 1) Абстрактный класс не может быть инстанцирован
- 2) Абстрактный класс не может содержать методов
- 3) Абстрактный класс должен содержать только абстрактные методы
- 4) Абстрактный класс может быть реализован в других классах

**17. Как в языке Java можно реализовать интерфейс?**

- 1) С помощью ключевого слова implements
- 2) С помощью ключевого слова interface
- 3) С помощью ключевого слова extends
- 4) С помощью ключевого слова override

**18. Что произойдет, если класс не переопределяет метод toString()?**

- 1) Будет вызвана стандартная реализация, возвращающая строковое представление объекта

- 2) Программа не скомпилируется
- 3) Метод не будет доступен в объекте
- 4) Программа завершится с ошибкой выполнения

**19. Какое из следующих утверждений относится к инкапсуляции?**

- 1) Соккрытие данных объекта
- 2) Способность класса наследовать другие классы
- 3) Использование абстракции
- 4) Перегрузка методов

**ПК-3**

**20. Что делает метод equals() в классе Object?**

- 1) Сравнивает ссылки на объекты
- 2) Сравнивает содержимое объектов
- 3) Вызывает метод hashCode()
- 4) Преобразует объект в строку

**21. Какой тип данных не является примитивным в языке Java?**

- 1) int
- 2) char
- 3) boolean
- 4) String

**22. Что делает метод hashCode() в классе Object?**

- 1) Возвращает строковое представление объекта
- 2) Генерирует уникальный идентификатор для объекта
- 3) Сравнивает объекты
- 4) Проверяет тип объекта

**23. Какой модификатор доступа позволяет методам быть доступными только внутри пакета?**

- 1) public
- 2) private
- 3) protected
- 4) default

**24. Какой принцип ООП позволяет создавать новые классы на основе существующих?**

- 1) Наследование
- 2) Инкапсуляция
- 3) Полиморфизм
- 4) Абстракция

**25. Что означает метод clone() в Java?**

- 1) Создает новый объект с идентичными данными
- 2) Копирует строку в новый объект
- 3) Переопределяет базовую реализацию toString()
- 4) Копирует объект в базу данных

**26. Как называется возможность классов изменять своё поведение в зависимости от типа объекта, на который они ссылаются?**

- 1) Полиморфизм
- 2) Абстракция
- 3) Наследование
- 4) Инкапсуляция

**27. Какой тип данных не может быть использован как тип параметра в дженериках?**

- 1) Object
- 2) int
- 3) String
- 4) List

**28. Что такое исключение в ООП?**

- 1) Ошибка в компиляции программы
- 2) Механизм обработки ошибок во время выполнения
- 3) Процесс вызова методов в классе
- 4) Видимость класса для других классов

**29. Что произойдет, если в классе будет вызван метод, который не определен в этом классе или его родителях?**

- 1) Компилятор выведет ошибку
- 2) Программа продолжит выполнение, но метод не будет работать
- 3) Метод будет выполнен с ошибкой
- 4) Процесс компиляции завершится успешно

**30. Как в Java обозначаются абстрактные методы?**

- 1) Метод с пустым телом
- 2) Метод с ключевым словом abstract
- 3) Метод с аннотацией @Override
- 4) Метод без имени

**31. Какой модификатор доступа позволяет членам класса быть доступными в пределах всего проекта?**

- 1) private
- 2) protected
- 3) public
- 4) default

**32. Какой тип наследования запрещен в языке Java?**

- 1) Множественное наследование
- 2) Наследование от абстрактных классов
- 3) Наследование интерфейсов
- 4) Наследование от классов

**33. Какой из принципов ООП требует, чтобы объект мог изменять только свои данные?**

- 1) Полиморфизм
- 2) Инкапсуляция
- 3) Наследование
- 4) Абстракция

**34. Какой из методов является обязательным для реализации интерфейса в Java?**

- 1) toString()
- 2) equals()
- 3) Любой метод интерфейса
- 4) hashCode()

**35. Что происходит, если в языке Java не определен метод finalize()?**

- 1) Программа не скомпилируется
- 2) Метод не будет вызван при удалении объекта
- 3) Компилятор автоматически генерирует этот метод
- 4) Программа завершится с ошибкой

**36. Как можно изменить содержимое final-переменной в Java?**

- 1) Присвоить новое значение
- 2) Невозможно изменить содержимое
- 3) Использовать метод set()
- 4) Переопределить метод

**37. Как можно проверить, является ли объект экземпляром определенного класса в Java?**

- 1) Использовать метод instanceof
- 2) Использовать метод equals()
- 3) Использовать метод ==
- 4) Использовать метод getClass()

**38. Какой из следующих принципов является основой интерфейсов в ООП?**

- 1) Полиморфизм
- 2) Инкапсуляция
- 3) Абстракция
- 4) Наследование

**39. Что такое событие в ООП?**

- 1) Способ передачи сообщений между объектами
- 2) Метод, вызываемый автоматически при изменении состояния объекта
- 3) Метод обработки исключений
- 4) Способ передачи данных через сетевые соединения

**40. Как называется метод, который вызывается при создании нового объекта?**

- 1) Destructor
- 2) Constructor
- 3) Finalize
- 4) Initialize

**41. Что произойдет, если попытаться вызвать абстрактный метод напрямую из объекта?**

- 1) Компилятор выдаст ошибку
- 2) Метод выполнится, но будет неэффективным
- 3) Метод выполнится с ошибкой
- 4) Программа завершится корректно

**42. Какое из свойств классов доступно в объектно-ориентированных языках программирования?**

- 1) Поля
- 2) Методы
- 3) Конструкторы
- 4) Все вышеуказанное

**43. Что такое интерфейс-метод?**

- 1) Метод, который не имеет реализации
- 2) Метод, доступный только в интерфейсах
- 3) Метод с одним аргументом
- 4) Метод, который нельзя переопределить

**44. Что происходит при использовании модификатора `static` в классе?**

- 1) Поле или метод становятся доступными без создания экземпляра класса
- 2) Поле или метод становятся доступными только в экземпляре класса
- 3) Поле или метод становятся скрытыми для других классов
- 4) Поле или метод могут быть изменены в любом месте

**45. Какой тип класса можно использовать для описания объектов, у которых нет состояния, но есть методы для выполнения определенной работы?**

- 1) Абстрактный класс

- 2) Интерфейс
- 3) Класс с методами и переменными
- 4) Класс-обработчик событий

**Ключ к тесту:**

1.2	2.2	3.2	4.1	5.2	6.1	7.1	8.3	9.2
10.2	11.2	12.2	13.4	14.2	15.1	16.1	17.1	18.1
19.1	20.2	21.4	22.2	23.4	24.1	25.1	26.1	27.2
28.2	29.1	30.2	31.3	32.1	33.2	34.3	35.2	36.2
37.1	38.3	39.2	40.2	41.1	42.4	43.1	44.1	45.2

### Шкала оценивания результатов тестирования

% верных решений (ответов)	Шкала оценивания
85 - 100	отлично
70 - 84	хорошо
50- 69	удовлетворительно
0 - 49	неудовлетворительно

#### 6.2.4. Примеры задач при разборе конкретных ситуаций

*Тема 1. Введение в объектно-ориентированное программирование.*

*Введение в объектно-ориентированное проектирование.*

##### **ОПК-8**

1. Написать программу на языке Java/C++, которая демонстрирует основные принципы объектно-ориентированного программирования: инкапсуляцию, наследование и полиморфизм. Программа должна включать классы, методы и объекты, а также примеры их взаимодействия.

##### **ПК-3**

2. Создать UML-диаграмму для описания системы управления библиотекой. Определить классы (например, Книга, Читатель, Библиотекарь) и их взаимосвязи. Описать, какие атрибуты и методы будут у каждого класса.

3. Провести анализ различных языков программирования (Java, C++, Python) с точки зрения их поддержки объектно-ориентированных принципов. Написать отчет о том, как каждый язык внедряет принципы ООП.  
*Тема 2. Классы и объекты. Работа с объектами*

##### **ОПК-8**

1. Реализовать класс "Круг" на языке Java/C++, который будет содержать методы для вычисления площади и периметра. Создать объект класса и вывести результаты на экран.

##### **ПК-3**

2. Написать программу, которая создает массив объектов класса "Студент". Каждый студент должен иметь имя, фамилию и баллы за экзамены. Реализовать метод, который будет вычислять средний балл для каждого студента.

3. Создать класс "Автомобиль" с атрибутами (марка, модель, год выпуска) и методами (вывод информации, изменение года выпуска). Написать пример использования этого класса и продемонстрировать создание объектов.

*Тема 3. Наследование. Одиночное и множественное наследование. Полиморфизм. Виртуальные методы*

### **ОПК-8**

1. Реализовать базовый класс "Животное" с виртуальным методом "издать звук". Создать два производных класса ("Собака" и "Кошка"), переопределяющих метод "издать звук". Написать программу, которая демонстрирует полиморфизм при использовании этих классов.

### **ПК-3**

2. Написать программу, которая реализует одиночное и множественное наследование. Создать классы "Форма" (базовый класс), "Круг" и "Квадрат" (производные классы) и "Треугольник" (класс, наследующий от "Форма" и реализующий дополнительный интерфейс).

3. Создать класс "Фигура" с методом "вычислить площадь". Реализовать несколько производных классов (например, "Круг", "Прямоугольник") и написать программу, которая принимает массив фигур и выводит их площади, демонстрируя полиморфизм.

*Тема 4. Обработка исключений. Шаблоны классов. Поточные классы. Контейнерные классы.*

### **ОПК-8**

1. Написать программу на языке Java/C++, которая обрабатывает исключения, возникающие при делении на ноль. Программа должна запрашивать у пользователя ввод чисел и корректно обрабатывать ошибочные вводы.

### **ПК-3**

2. Создать шаблон класса "Массив", который будет поддерживать операции добавления, удаления и доступа к элементам. Реализовать обработку исключений для случаев, когда происходит выход за пределы массива.

3. Написать программу, которая использует потоковые классы для чтения данных из текстового файла и записи их в другой файл. Обработать возможные исключения, такие как "файл не найден" и "нет доступа к файлу".

## **Шкала оценивания**

Шкала оценивания	Критерии оценивания
«Отлично»	обучающийся ясно изложил условие задачи, решение обосновал
«Хорошо»	обучающийся ясно изложил условие задачи, но в обосновании решения имеются сомнения;

«Удовлетворительно»	обучающийся изложил решение задачи, но обосновал его формулировками обыденного мышления;
«Неудовлетворительно»	обучающийся не уяснил условие задачи, решение не обосновал либо не сдал работу на проверку (в случае проведения решения задач в письменной форме).

### 6.2.5. Темы для рефератов

Тема (раздел)	Вопросы
Тема 1. Введение в объектно-ориентированное программирование. Введение в объектно-ориентированное проектирование.	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. История и эволюция объектно-ориентированного программирования.</li> <li>2. Основные принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм.</li> <li>3. Сравнение объектно-ориентированного программирования с процедурным и функциональным программированием.</li> <li>4. Объектно-ориентированное проектирование: понятие и основные этапы.</li> <li>5. UML (Unified Modeling Language) в объектно-ориентированном проектировании: структура и применение.</li> </ol> <p>ПК-3</p> <ol style="list-style-type: none"> <li>6. Паттерны проектирования: определение и примеры использования в ООП.</li> <li>7. Роль абстракции в объектно-ориентированном программировании.</li> <li>8. Применение ООП в современных языках программирования (Java, C++, Python и др.).</li> <li>9. Объектно-ориентированное программирование в веб-разработке: подходы и технологии.</li> <li>10. Будущее объектно-ориентированного программирования: новые тренды и направления.</li> </ol>
Тема 2. Классы и объекты. Работа с объектами	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Определение и структура классов и объектов в объектно-ориентированном программировании.</li> <li>2. Методы и атрибуты классов: различия и примеры использования.</li> <li>3. Конструкторы и деструкторы: определение, использование и значение.</li> <li>4. Инкапсуляция данных: модификаторы доступа и их роль в безопасности кода.</li> <li>5. Пример реализации классов и объектов на языке Java/C++/Python.</li> </ol> <p>ПК-3</p> <ol style="list-style-type: none"> <li>6. Взаимодействие объектов: методы и сообщения.</li> <li>7. Статические и динамические методы в классах: различия и примеры.</li> <li>8. Область видимости переменных и методов: локальные и глобальные.</li> <li>9. Применение объектов в реальных задачах: примеры из практики.</li> <li>10. Ошибки и проблемы, возникающие при работе с классами и</li> </ol>

	объектами.
Тема 3. Наследование. Одиночное и множественное наследование. Полиморфизм. Виртуальные методы	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Определение и принципы наследования в объектно-ориентированном программировании.</li> <li>2. Одиночное и множественное наследование: преимущества и недостатки.</li> <li>3. Полиморфизм в ООП: понятие и примеры применения.</li> <li>4. Виртуальные методы и их роль в реализации полиморфизма.</li> <li>5. Примеры использования наследования в реальных проектах.</li> </ol> <p>ПК-3</p> <ol style="list-style-type: none"> <li>6. Обработка проблем, связанных с множественным наследованием: алмазная проблема.</li> <li>7. Паттерны проектирования, основанные на наследовании: примеры и объяснения.</li> <li>8. Виртуальные и абстрактные классы: различия и применение.</li> <li>9. Роль интерфейсов в объектно-ориентированном программировании.</li> <li>10. Будущее наследования и полиморфизма в современных языках программирования.</li> </ol>
Тема 4. Обработка исключений. Шаблоны классов. Потокные классы. Контейнерные классы.	<p>ОПК-8</p> <ol style="list-style-type: none"> <li>1. Определение и значение обработки исключений в объектно-ориентированном программировании.</li> <li>2. Механизмы обработки исключений: try, catch, finally и их применение.</li> <li>3. Создание пользовательских исключений в ООП: примеры и рекомендации.</li> <li>4. Шаблоны классов: определение и примеры использования в практических задачах.</li> <li>5. Потокные классы: работа с потоками ввода-вывода в ООП.</li> </ol> <p>ПК-3</p> <ol style="list-style-type: none"> <li>6. Контейнерные классы: определение, примеры и использование в программировании.</li> <li>7. Сравнение различных контейнерных классов в Java/C++ (например, списки, множества, карты).</li> <li>8. Применение потокных классов в многопоточном программировании.</li> <li>9. Проблемы, связанные с обработкой исключений и работой с потоками.</li> <li>10. Будущее обработки исключений и контейнерных классов в современных языках программирования.</li> </ol>

### Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему самостоятельной работы, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает тему самостоятельной работы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.

«Удовлетворительно»	Обучающийся в целом раскрывает тему самостоятельной работы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой самостоятельной работы

## 6.2. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

**Вопросы для подготовки к промежуточной аттестации по итогам освоения дисциплины Объектно-ориентированное программирование:**

### **ОПК-8.**

1. Эволюция методологий программирования. Парадигмы программирования.
2. Основные принципы объектно-ориентированного программирования: абстрагирование, инкапсуляция, модульность, иерархия, типизация, параллелизм, сохраняемость.
3. Объект с точки зрения ООП. Состояние. Поведение. Идентичность и жизненный цикл объектов. Взаимоотношения между объектами.
4. Понятие объектно-ориентированного проектирования.
5. Язык UML (понятие, определение, мотивация применения, типы основных диаграмм).
6. Язык UML. Диаграмма вариантов использования. Документ «Поток событий».
7. Язык UML. Диаграмма последовательности.
8. Язык UML. Кооперативная диаграмма.
9. Язык UML. Отношения между классами (обобщения, ассоциации, зависимости и реализации).
10. Язык UML. Диаграмма классов. Диаграмма пакетов.
11. Язык UML. Диаграмма состояний. Диаграмма деятельности.
12. Язык UML. Диаграмма компонентов. Диаграмма размещения.
13. Объявление класса. Методы inline.
14. Конструктор (назначение, свойства, перегрузка, вызов).
15. Конструктор копирования.
16. Статические поля и методы класса.
17. Деструктор.
18. Дружественные функции и классы.
19. Перегруженные операторы и методы класса.
20. Простое наследование. Механизм наследования.
21. Переопределение методов и виртуальные функции.
22. Многократное наследование.
23. Конструкторы и деструкторы при наследовании.
24. Чисто виртуальные методы и абстрактные классы.

25. Шаблоны классов. Типы по умолчанию и явная специализация обобщённого класса.
26. Разделение объектно-ориентированной программы на потоки. Изучение способов синхронизации потоков в C++.
27. Обработка исключений в C++(механизм, синтаксис, перехват).
28. Исключения в конструкторах и деструкторах.
29. Классы стандартной библиотеки C++. Поточковые классы. Стандартные потоки. Форматирование данных.
30. Поточковые классы: ошибки потоков.
31. Поточковые классы : файловые потоки (конструкторы, режимы открытия, закрытие).
32. Класс string стандартной библиотеки (конструкторы, операции, функции).
33. Контейнеры. Виды контейнеров. Итераторы.
34. Что такое инкапсуляция и как она реализуется в объектно-ориентированных языках программирования?
35. Что такое абстракция и как она помогает при проектировании сложных систем

### **ПК-3**

36. В чем отличие между классами и объектами в ООП?
37. Что такое интерфейс и как он отличается от абстрактного класса?
38. Ожидаемое поведение методов в объектно-ориентированном программировании.
39. Разница между public, private и protected модификаторами доступа в ООП.
40. Преимущества и недостатки использования наследования в ООП.
41. Как можно применить полиморфизм в реальных проектах?
42. Механизм позднего связывания и его использование.
43. Как работают исключения в ООП и какие типы исключений могут быть в языке программирования C++?
44. Что такое интерфейсы и абстрактные классы в C++?
45. Как можно реализовать шаблонный метод (Template Method) в ООП?
46. Пример реализации и использования контейнеров в ООП.
47. Применение итераторов для работы с контейнерами.
48. Операции с потоками ввода/вывода (I/O) в ООП.
49. Работа с файлами и потоки данных в C++.
50. Что такое стек и очередь, как их можно реализовать в ООП?
51. Алгоритмы сортировки и поиска: реализация в ООП.
52. Что такое принцип единой ответственности (SRP) в объектно-ориентированном проектировании?
53. Что такое принцип открытости/закрытости (OCP) в ООП?
54. Как реализовать паттерн проектирования "Стратегия" в ООП?
55. Паттерн проектирования "Одиночка" (Singleton): принцип и реализация.

56. Паттерн проектирования "Фабрика" (Factory Method): что это такое и когда его использовать?
57. Паттерн проектирования "Наблюдатель" (Observer): принцип и пример реализации.
58. Паттерн проектирования "Команда" (Command): когда и как использовать.
59. Паттерн проектирования "Декоратор" (Decorator): особенности и использование.
60. Паттерн проектирования "Адаптер" (Adapter): как он помогает при интеграции разных интерфейсов.
61. Описание принципов SOLID в ООП.
62. Основные паттерны проектирования в C++.
63. Описание структуры данных "Двоичное дерево" и его применение.
64. Взаимодействие объектов в многозадачных и многопоточных приложениях.
65. Принципы проектирования интерфейсов в объектно-ориентированном программировании.
66. Тестирование объектов и классов в ООП: основы юнит-тестирования.
67. Использование макросов и шаблонов в C++ для оптимизации кода.
68. Преимущества и ограничения работы с динамическим и статическим полиморфизмом.
69. Различие между типами данных value и reference в ООП.
70. Понятие и использование дескрипторов классов в Python (или другом языке ООП).
71. Применение паттерна "Фабричный метод" в реальных проектах.
72. Роль рефлексии в объектно-ориентированном программировании.
73. Как избежать проблемы глубоких копий в ООП при передаче объектов?
74. Методы улучшения производительности ООП-программ, основанных на контейнерах и потоках.
75. Различие между коллекциями и контейнерами в ООП, их преимущества и области применения.

#### **6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Основной целью проведения промежуточной аттестации является определение степени достижения целей по учебной дисциплине или ее разделам. Осуществляется это проверкой и оценкой уровня теоретической знаний, полученных обучающимися, умения применять их в решении практических задач, степени овладения обучающимися практическими

навыками и умениями в объеме требований рабочей программы по дисциплине, а также их умение самостоятельно работать с учебной литературой.

Организация проведения промежуточной аттестации регламентирована «Положением об организации образовательного процесса в федеральном государственном автономном образовательном учреждении «Московский политехнический университет»

#### 6.4.1. Показатели оценивания компетенций на различных этапах их формирования, достижение обучающимися планируемых результатов обучения по дисциплине

<b>Код и наименование компетенции ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения</b>				
<b>Этап (уровень)</b>	<b>Критерии оценивания</b>			
	<b>неудовлетворительно</b>	<b>удовлетворительно</b>	<b>хорошо</b>	<b>отлично</b>
<b>знать</b>	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: основных языков программирования, операционных систем и оболочек, современных сред разработки программного обеспечения	Обучающийся демонстрирует неполное соответствие следующих знаний: основных языков программирования, операционных систем и оболочек, современных сред разработки программного обеспечения	Обучающийся демонстрирует частичное соответствие следующих знаний: основных языков программирования, операционных систем и оболочек, современных сред разработки программного обеспечения	Обучающийся демонстрирует полное соответствие следующих знаний: основных языков программирования, операционных систем и оболочек, современных сред разработки программного обеспечения
<b>уметь</b>	Обучающийся не умеет или в недостаточной степени умеет составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули	Обучающийся демонстрирует неполное соответствие следующих умений: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули	Обучающийся демонстрирует частичное соответствие следующих умений: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули	Обучающийся демонстрирует полное соответствие следующих умений: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули

<b>владеть</b>	Обучающийся не владеет или в недостаточной степени владеет языком программирования, методами отладки и тестирования работоспособности программы	Обучающийся владеет в неполном объеме и проявляет недостаточность владения навыками программирования, методами отладки и тестирования работоспособности программы	Обучающимся допускаются незначительные ошибки, неточности, затруднения, частично владеет навыками программирования, методами отладки и тестирования работоспособности программы	Обучающийся свободно применяет полученные навыки, в полном объеме владеет программированием, методами отладки и тестирования работоспособности программы
<b>Код и наименование компетенции ПК-3. Способен разрабатывать компоненты системных программных продуктов</b>				
<b>Этап (уровень)</b>	<b>Критерии оценивания</b>			
	<b>неудовлетворительно</b>	<b>удовлетворительно</b>	<b>хорошо</b>	<b>отлично</b>
<b>знать</b>	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: основы системного администрирования, современных систем управления базами данных, устройство и функционирование современных ИС	Обучающийся демонстрирует неполное соответствие следующих знаний: основы системного администрирования, современных систем управления базами данных, устройство и функционирование современных ИС	Обучающийся демонстрирует частичное соответствие следующих знаний: основы системного администрирования, современных систем управления базами данных, устройство и функционирование современных ИС	Обучающийся демонстрирует полное соответствие следующих знаний: основы системного администрирования, современных систем управления базами данных, устройство и функционирование современных ИС
<b>уметь</b>	Обучающийся не умеет или в недостаточной степени умеет выполнять устанавливать и настраивать СУБД, прикладное ПО	Обучающийся демонстрирует неполное соответствие следующих умений: устанавливать и настраивать СУБД, прикладное ПО	Обучающийся демонстрирует частичное соответствие следующих умений: устанавливать и настраивать СУБД, прикладное ПО	Обучающийся демонстрирует полное соответствие следующих умений: устанавливать и настраивать СУБД, прикладное ПО

<b>владеть</b>	Обучающийся владеет или не владеет в недостаточной степени настройкой прикладного ПО, необходимого для функционирования ИС, настройкой СУБД для оптимального функционирования ИС	Обучающийся владеет в неполном объеме и проявляет недостаточность владения навыками настройкой прикладного ПО, необходимого для функционирования ИС, настройкой СУБД для оптимального функционирования ИС	Обучающимся допускаются незначительные ошибки, неточности, затруднения, частично владеет навыками настройкой прикладного ПО, необходимого для функционирования ИС, настройкой СУБД для оптимального функционирования ИС	Обучающийся свободно применяет полученные навыки, в полном объеме владеет настройкой прикладного ПО, необходимого для функционирования ИС, настройкой СУБД для оптимального функционирования ИС
----------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### 6.4.2. Методика оценивания результатов промежуточной аттестации

Показателями оценивания компетенций на этапе промежуточной аттестации по дисциплине «Объектно-ориентированное программирование» являются результаты обучения по дисциплине.

#### Оценочный лист результатов обучения по дисциплине

Код компетенции	Знания	Умения	Навыки	Уровень сформированности компетенции на данном этапе / оценка
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	на уровне знаний: знать основных языков программирования, операционных систем и оболочек, современных сред разработки программного обеспечения	на уровне умений: уметь составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать	на уровне навыков: владеет языком программирования, методами отладки и тестирования работоспособности и программы	

		ь программные модули		
ПК-3. Способен разрабатывать компоненты системных программных продуктов	на уровне знаний: знать основы системного администрирования, современных систем управления базами данных, устройство и функционирование современных ИС	на уровне умений: уметь устанавливать и настраивать СУБД, прикладное ПО	на уровне навыков: настройкой прикладного ПО, необходимого для функционирования ИС, настройкой СУБД для оптимального функционирования ИС	
Оценка по дисциплине (среднее арифметическое)				

Оценка по дисциплине зависит от уровня сформированности компетенций, закрепленных за дисциплиной, и представляет собой среднее арифметическое от выставленных оценок по отдельным результатам обучения (знания, умения, навыки).

Оценка «отлично» выставляется, если среднее арифметическое находится в интервале от 4,5 до 5,0.

Оценка «хорошо» выставляется, если среднее арифметическое находится в интервале от 3,5 до 4,4.

Оценка «удовлетворительно» выставляется, если среднее арифметическое находится в интервале от 2,5 до 3,4.

Оценка «неудовлетворительно» выставляется, если среднее арифметическое находится в интервале от 0 до 2,4.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по дисциплине «Объектно-ориентированное программирование», при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем, ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Шкала оценивания	Описание
------------------	----------

Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

## 7. Электронная информационно-образовательная среда

Каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронной информационно-образовательной среде Чебоксарского института (филиала) Московского политехнического университета из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), как на территории филиала, так и вне ее.

Электронная информационно-образовательная среда – совокупность информационных и телекоммуникационных технологий, соответствующих технологических средств, обеспечивающих освоение обучающимися образовательных программ в полном объеме независимо от места нахождения обучающихся.

Электронная информационно-образовательная среда обеспечивает:

а) доступ к учебным планам, рабочим программам дисциплин (модулей), практик, электронным учебным изданиям и электронным образовательным ресурсам, указанным в рабочих программах дисциплин (модулей), практик;

б) формирование электронного портфолио обучающегося, в том числе сохранение его работ и оценок за эти работы;

в) фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения программы бакалавриата;

г) проведение учебных занятий, процедур оценки результатов обучения, реализация которых предусмотрена с применением электронного обучения, дистанционных образовательных технологий;

д) взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействия посредством сети "Интернет".

Функционирование электронной информационно-образовательной среды обеспечивается соответствующими средствами информационно-коммуникационных технологий и квалификацией работников, ее использующих и поддерживающих.

Функционирование электронной информационно-образовательной среды соответствует законодательству Российской Федерации.

Основными составляющими ЭИОС филиала являются:

а) сайт института в сети Интернет, расположенный по адресу [www.polytech21.ru](http://www.polytech21.ru), <https://chebpolytech.ru/> который обеспечивает:

- доступ обучающихся к учебным планам, рабочим программам дисциплин, практик, к изданиям электронных библиотечных систем, электронным информационным и образовательным ресурсам, указанных в рабочих программах (разделы сайта «Сведения об образовательной организации»);

- информирование обучающихся обо всех изменениях учебного процесса (новостная лента сайта, лента анонсов);

- взаимодействие между участниками образовательного процесса (подразделы сайта «Задать вопрос директору»);

б) официальные электронные адреса подразделений и сотрудников института с Яндекс-доменом [@polytech21.ru](mailto:@polytech21.ru) (список контактных данных подразделений Филиала размещен на официальном сайте Филиала в разделе «Контакты», списки контактных официальных электронных данных преподавателей размещены в подразделах «Кафедры») обеспечивают взаимодействие между участниками образовательного процесса;

в) личный кабинет обучающегося (портфолио) (вход в личный кабинет размещен на официальном сайте Филиала в разделе «Студенту» подразделе «Электронная информационно-образовательная среда») включает в себя портфолио студента, электронные ведомости, рейтинг студентов и обеспечивает:

- фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ обучающимися,

- формирование электронного портфолио обучающегося, в том числе с сохранение работ обучающегося, рецензий и оценок на эти работы,

г) электронные библиотеки, включающие электронные каталоги, полнотекстовые документы и обеспечивающие доступ к учебно-методическим материалам, выпускным квалификационным работам и т.д.:

Чебоксарского института (филиала) - «ИРБИС»

д) электронно-библиотечные системы (ЭБС), включающие электронный каталог и полнотекстовые документы:

- ЭБС «ЛАНЬ» -<https://e.lanbook.com/>
- Образовательная платформа Юрайт - <https://urait.ru>
- IPR SMART -<https://www.iprbookshop.ru/>

е) платформа цифрового образования Политеха - <https://lms.mospolytech.ru/>

ж) система «Антиплагиат» -<https://www.antiplagiat.ru/>

з) система электронного документооборота DIRECTUM Standard — обеспечивает документооборот между Филиалом и Университетом;

и) система «IC Управление ВУЗом Электронный деканат» (Московский политехнический университет) обеспечивает фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ обучающимися;

к) система «POLYTECH systems» обеспечивает информационное, документальное автоматизированное сопровождение образовательного процесса;

л) система «Абитуриент» обеспечивает документальное автоматизированное сопровождение работы приемной комиссии.

## **8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

### Основная литература:

1. Тузовский, А. Ф. Объектно-ориентированное программирование : учебник для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 213 с. — (Высшее образование). — ISBN 978-5-534-16316-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561394>.

2. Зыков, С. В. Объектно-ориентированное программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 151 с. — (Высшее образование). — ISBN 978-5-534-16941-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561434>.

3. Свердлов, С. З. Языки программирования и методы трансляции : учебное пособие для вузов / С. З. Свердлов. — 5-е изд., стер. — Санкт-Петербург : Лань, 2025. — 564 с. — ISBN 978-5-507-50570-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/447398> (дата обращения: 20.05.2025). — Режим доступа: для авториз. пользователей.

### Дополнительная литература:

1. Казанский, А. А. Объектно-ориентированное программирование. Visual Basic : учебник для среднего профессионального образования / А. А. Казанский. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 295 с. — (Профессиональное образование). — ISBN 978-5-534-21384-3. — Текст :

электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/569868>.

2. Программирование. Сборник задач : учебное пособие для вузов / О. Г. Архипов, В. С. Батасова, П. В. Гречкина [и др.] ; под редакцией М. М. Маран. — 3-е изд., стер. — Санкт-Петербург : Лань, 2025. — 140 с. — ISBN 978-5-507-50516-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/443291>. — Режим доступа: для авториз. пользователей.

#### Периодика:

Системы управления и информационные технологии: научный журнал - URL: <http://sbook.ru/suit/>- Текст : электронный

### **9. Профессиональные базы данных и информационно-справочные системы**

Профессиональная база данных и информационно-справочные системы	Информация о праве собственности (реквизиты договора)
Университетская информационная система РОССИЯ <a href="https://uisrussia.msu.ru/">https://uisrussia.msu.ru/</a>	Тематическая электронная библиотека и база для прикладных исследований в области экономики, управления, социологии, лингвистики, философии, филологии, международных отношений, права. свободный доступ
научная электронная библиотека Elibrary <a href="http://elibrary.ru/">http://elibrary.ru/</a>	Научная электронная библиотека eLIBRARY.RU - это крупнейший российский информационно-аналитический портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты более 26 млн научных статей и публикаций, в том числе электронные версии более 5600 российских научно-технических журналов, из которых более 4800 журналов в открытом доступе свободный доступ
сайт Института научной информации по общественным наукам РАН. <a href="http://www.inion.ru">http://www.inion.ru</a>	Библиографические базы данных ИНИОН РАН по социальным и гуманитарным наукам ведутся с начала 1980-х годов. Общий объем массивов составляет более 3 млн. 500 тыс. записей (данные на 1 января 2012 г.). Ежегодный прирост — около 100 тыс. записей. В базы данных включаются аннотированные описания книг и статей из журналов и сборников на 140 языках, поступивших в Фундаментальную библиотеку ИНИОН РАН. Описания статей и книг в базах данных снабжены шифром хранения и ссылками на полные тексты источников из Научной электронной библиотеки.
Федеральный портал «Российское образование» [Электронный ресурс] – <a href="http://www.edu.ru">http://www.edu.ru</a>	Федеральный портал «Российское образование» – уникальный интернет-ресурс в сфере образования и науки. Ежедневно публикует самые актуальные новости, анонсы событий, информационные материалы для широкого круга читателей. Ежедневно на портале размещаются эксклюзивные материалы, интервью с ведущими специалистами – педагогами, психологами, учеными, репортажи и аналитические статьи. Читатели получают доступ к нормативно-правовой базе сферы образования, они могут пользоваться самыми различными полезными сервисами – такими, как онлайн-тестирование, опросы по актуальным темам и т.д.

Информационные технологии – периодическое научно-техническое издание в области информационных технологий, автоматизированных систем и использования информатики в различных приложениях <a href="http://novtex.ru">novtex.ru</a>	Издательство выпускает теоретические и прикладные научно-технические журналы, обеспечивающие научной, производственной, обзорно-аналитической и образовательной информацией руководящих работников и специалистов промышленных предприятий, научных академических и отраслевых организаций, а также учебных заведений в области приоритетных направлений развития науки и технологий.
Ассоциация инженерного образования России <a href="http://www.ac-raee.ru/">http://www.ac-raee.ru/</a>	Совершенствование образования и инженерной деятельности во всех их проявлениях, относящихся к учебному, научному и технологическому направлениям, включая процессы преподавания, консультирования, исследования, разработки инженерных решений, оказания широкого спектра образовательных услуг, обеспечения связей с общественностью, производством, наукой и интеграции в международное научно-образовательное пространство. свободный доступ

### 10. Программное обеспечение (лицензионное и свободно распространяемое), используемое при осуществлении образовательного процесса

Аудитория	Программное обеспечение	Информация о праве собственности (реквизиты договора, номер лицензии и т.д.)
<b>№ 2196</b> Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой бакалавриата/специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Кабинет систем управления ООО «НПО «Каскад-ГРУП»	Windows 7 OLPNLAcdmc	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор № ППИ-126/2023 от 14.12.2023
	Google Chrome	Свободное распространяемое программное обеспечение (бессрочная лицензия)
	Zoom	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Microsoft Office Standard 2007(Microsoft DreamSpark Premium Electronic Software Delivery Academic(Microsoft Open License	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
<b>№ 2116</b> Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего	Windows 7 OLPNLAcdmc	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)

<p>профессионального образования/бакалавриата/специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Лаборатория «Программного обеспечения и сопровождения компьютерных систем» Кабинет информационных систем и технологий АО «НПК «ЭЛАРА»</p>	<p>Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License</p>	<p>Сублицензионный договор № ППИ-126/2023 от 14.12.2023</p>
	<p>Microsoft Visual Studio 2019</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>
	<p>КОМПАС-3D v20 и v21</p>	<p>Сублицензионный договор № Нп-22-00044 от 21.03.2022 (бессрочная лицензия)</p>
	<p>PaitNet</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>
	<p>AIMP</p>	<p>отечественное свободно распространяемое программное обеспечение (бессрочная лицензия)</p>
<p><b>№ 1126</b> Помещение для самостоятельной работы обучающихся</p>	<p>Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License</p>	<p>Сублицензионный договор № ППИ-126/2023 от 14.12.2023</p>
	<p>Windows 7 OLPNLAcDmc</p>	<p>договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)</p>
	<p>AdobeReader</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>
	<p>Гарант- справочно-правовая система</p>	<p>Договор №С-002-2025 от 09.01.2025</p>
	<p>Yandex браузер</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>
	<p>Microsoft Office Standard 2007(Microsoft DreamSpark Premium Electronic Software Delivery Academic(Microsoft Open License</p>	<p>номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)</p>
	<p>Zoom</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>
<p>AIMP</p>	<p>отечественное свободно распространяемое программное</p>	

		обеспечение (бессрочная лицензия)
--	--	-----------------------------------

## 11. Материально-техническое обеспечение дисциплины

Наименование объекта, подтверждающего наличие материально-технического обеспечения, с перечнем основного оборудования	Адрес (местоположение) объекта подтверждающего наличие МТО
Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой бакалавриата/ специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Кабинет систем управления ООО «НПО «Каскад-ГРУП» <b>2196</b> (Чебоксары, ул. К.Маркса, д.60)	<u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; стенды <u>Технические средства обучения:</u> мультимедийное оборудование (проектор, экран)
Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/ специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Лаборатория «Программного обеспечения и сопровождения компьютерных систем» Кабинет информационных систем и технологий АО «НПК «ЭЛАРА» <b>№ 2116</b> (г. Чебоксары, ул. К.Маркса, д.60)	<u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; стенды, автоматизированные рабочие места на 15 обучающихся, автоматизированное рабочее место преподавателя, <u>Технические средства обучения:</u> компьютерная техника; мультимедийное оборудование (проектор, экран), маркерная доска, программное обеспечение общего и профессионального назначения
Помещение для самостоятельной работы обучающихся <b>№ 1126</b> (г. Чебоксары, ул. К.Маркса. 60)	<u>Оборудование:</u> комплект мебели для учебного процесса; <u>Технические средства обучения:</u> компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Филиала

## 12. Методические указания для обучающегося по освоению дисциплины

### *Методические указания для занятий лекционного типа*

В ходе лекционных занятий обучающемуся необходимо вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Необходимо задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Целесообразно дорабатывать свой конспект лекции, делая в нем соответствующие записи из основной и дополнительной литературы, рекомендованной преподавателем и предусмотренной учебной программой дисциплины.

#### ***Методические указания для занятий лабораторного типа.***

Выполнению лабораторных работ предшествует проверка знаний студентов – их теоретической готовности к выполнению задания. Проверка знаний проводится в форме, которую определяет преподаватель дисциплины (тестирование, опрос).

При проведении лабораторных занятий выделяют следующие разделы:

- общие положения (перечень лабораторных или практических занятий);
- общие требования к выполнению работ;
- инструкция по каждой работе;
- справочные материалы и т. д.

Лабораторные занятия позволяют развивать у обучающегося творческое теоретическое мышление, умение самостоятельно изучать литературу, анализировать практику; учат четко формулировать мысль, вести дискуссию, то есть имеют исключительно важное значение в развитии самостоятельного мышления.

Готовясь к докладу или выступлению в рамках интерактивной формы, при необходимости, следует обратиться за помощью к преподавателю.

#### ***Методические указания к самостоятельной работе.***

Самостоятельная работа обучающегося является основным средством овладения учебным материалом во время, свободное от обязательных учебных занятий. Самостоятельная работа обучающегося над усвоением учебного материала по учебной дисциплине может выполняться в библиотеке университета, учебных кабинетах, компьютерных классах, а также в домашних условиях. Содержание и количество самостоятельной работы обучающегося определяется учебной программой дисциплины, методическими материалами, практическими заданиями и указаниями преподавателя.

#### ***Самостоятельная работа в аудиторное время может включать:***

- 1) конспектирование (составление тезисов) лекций;
- 2) выполнение контрольных работ;
- 3) решение задач;
- 4) работу со справочной и методической литературой;
- 5) работу с нормативными правовыми актами;
- 6) выступления с докладами, сообщениями на семинарских занятиях;
- 7) защиту выполненных работ;

8) участие в оперативном (текущем) опросе по отдельным темам изучаемой дисциплины;

9) участие в беседах, деловых (ролевых) играх, дискуссиях, круглых столах, конференциях;

10) участие в тестировании и др.

***Самостоятельная работа во внеаудиторное время может состоять из:***

1) повторения лекционного материала;

2) подготовки к практическим занятиям;

3) изучения учебной и научной литературы;

4) изучения нормативных правовых актов (в т.ч. в электронных базах данных);

5) решения задач, и иных практических заданий

6) подготовки к контрольным работам, тестированию и т.д.;

7) подготовки к практическим занятиям устных докладов (сообщений);

8) подготовки рефератов, эссе и иных индивидуальных письменных работ по заданию преподавателя;

9) выполнения курсовых работ, предусмотренных учебным планом;

10) выполнения выпускных квалификационных работ и др.

11) выделения наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями на консультациях.

12) проведения самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов, написания рефератов и эссе по отдельным вопросам изучаемой темы.

Текущий контроль осуществляется в форме устных, тестовых опросов, докладов, творческих заданий.

В случае пропусков занятий, наличия индивидуального графика обучения и для закрепления практических навыков студентам могут быть выданы типовые индивидуальные задания, которые должны быть сданы в установленный преподавателем срок.

### **13. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья**

Обучение по дисциплине «Объектно-ориентированное программирование» инвалидов и лиц с ограниченными возможностями здоровья (далее ОВЗ) осуществляется преподавателем с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Для студентов с нарушениями опорно-двигательной функции и с ОВЗ по слуху предусматривается сопровождение лекций и практических занятий мультимедийными средствами, раздаточным материалом.

Для студентов с ОВЗ по зрению предусматривается применение технических средств усиления остаточного зрения, а также предусмотрена возможность разработки аудиоматериалов.

По дисциплине «Объектно-ориентированное программирование» обучение инвалидов и лиц с ограниченными возможностями здоровья может осуществляться как в аудитории, так и с использованием электронной информационно-образовательной среды, образовательного портала и электронной почты.

## ЛИСТ ДОПОЛНЕНИЙ И ИЗМЕНЕНИЙ

### рабочей программы дисциплины

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202\_\_-202\_\_ учебном году на заседании кафедры, протокол № \_\_\_ от « » 202 г.

Внесены дополнения и изменения \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202\_\_-202\_\_ учебном году на заседании кафедры, протокол № \_\_\_ от « » 202 г.

Внесены дополнения и изменения \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202\_\_-202\_\_ учебном году на заседании кафедры, протокол № \_\_\_ от « » 202 г.

Внесены дополнения и изменения \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202\_\_-202\_\_ учебном году на заседании кафедры, протокол № \_\_\_ от « » 202 г.

Внесены дополнения и изменения \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_