Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 04.10.2023 16:37:51
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет машиностроения

УТВЕРЖДАЮ

Декан

_____ /Е.В. Сафонов/

«27» апреля 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Объектно-ориентированное программирование

Направление подготовки
11.03.01 Радиотехника

Профиль
Интеллектуальная радиоэлектроника и промышленный интернет вещей

Квалификация
Бакалавр

Формы обучения
очная

Москва, 2023 г.

Разработчик(и):

Старший преподаватель кафедры АиУ  /Т.А. Лисовская/

Согласовано:
Заведующий кафедрой «Автоматика и управление»,
д.т.н., профессор  /А.А. Радионов/

Руководитель образовательной программы
д.т.н., профессор  /А.А. Радионов/

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 4
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость 5
3.2 Тематический план изучения дисциплины 5
3.3 Содержание дисциплины 6
3.4 Тематика семинарских/практических и лабораторных занятий 6
3.5 Тематика курсовых проектов (курсовых работ) 6
4 Учебно-методическое и информационное обеспечение 7
4.1 Нормативные документы и ГОСТы 7
4.2 Основная литература 7
4.3 Дополнительная литература 7
4.4 Электронные образовательные ресурсы 7
4.5 Лицензионное и свободно распространяемое программное обеспечение 7
4.6 Современные профессиональные базы данных и информационные справочные системы 8
5 Материально-техническое обеспечение 8
6 Методические рекомендации 8
6.1 Методические рекомендации для преподавателя по организации обучения 8
6.2 Методические указания для обучающихся по освоению дисциплины 9
7 Фонд оценочных средств 9
7.1 Методы контроля и оценивания результатов обучения 10
7.2 Шкала и критерии оценивания результатов обучения 11
7.3 Оценочные средства 15

1 Цели, задачи и планируемые результаты обучения по дисциплине

Целью изучения дисциплины "Объектно-ориентированное программирование" состоит в развитии у студентов базовых навыков программирования на языке высокого уровня (Python) в объектно-ориентированной парадигме программирования, создания объектно-ориентированных программ и алгоритмизации.

Задачи изучения дисциплины:

- Формирование систематизированного представления о концепциях, моделях и принципах организации в объектно-ориентированной парадигме;
- Ознакомление с современным состоянием и перспективными направлениями развития программирования;
- Выработка практических навыков в области выбора и применения технологий программирования.

Обучение по дисциплине «Объектно-ориентированное программирование» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции	Наименование показателя оценивания
ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ИОПК-5.1 Понимает методы алгоритмизации, языки и технологии программирования; ИОПК-5.2 Применяет методы алгоритмизации, языки и технологии программирования при решении профессиональных задач; ИОПК-5.2 Разрабатывает алгоритмы и компьютерные программы, пригодные для практического применения	**Знать:** Методы разработки алгоритмов в парадигме объектно-ориентированного программирования **Уметь:** Описывать задачу с точки зрения объектно-ориентированного подхода и подбирать соответствующие структуры для разработки алгоритма и программного кода **Владеть:** Навыками программирования на языке высокого уровня и алгоритмизации в объектно-ориентированной парадигме программирования

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части блока Б1 «Дисциплины (модули)».

Дисциплина непосредственно связана со следующими дисциплинами и практиками ООП:

Интеллектуальный анализ данных;
Информационные технологии;
Компьютерное зрение;
Машинное обучение;
Основы алгоритмизации и программирования;
Цифровая грамотность.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы (144 часа).

3.1 Виды учебной работы и трудоемкость

№ п/п	Вид учебной работы	Количество часов	Семестр
			4
1	**Аудиторные занятия**	**54**	**54**
	В том числе:		
1.1	Лекции	18	18
1.2	Семинарские/практические занятия	36	36
2	**Самостоятельная работа**	**90**	**90**
	В том числе:		
2.1	Подготовка отчётов по практическим работам	40	40
2.2	Работа с конспектом лекций	30	30
2.3	Подготовка к диф.зачёту	20	20
3	**Промежуточная аттестация**		
	Зачет/диф.зачет/экзамен		диф.зачет
	Итого	**144**	**144**

3.2 Тематический план изучения дисциплины

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/практические занятия	Лабораторные занятия	Практическая подготовка	
1	**Раздел 1. Введение в объектно-ориентированное программирование**	**71**	**8**	**18**	**0**	**0**	**45**
1.1	Тема 1. Объектно-ориентированная парадигма программирования. Основные понятия, определения, преимущества и недостатки		2	2			10
1.2	Тема 2. Основы классов и объектов		2	6			15
1.3	Тема 3. Наследование и полиморфизм		2	6			10
1.4	Тема 4. Инкапсуляция и защита данных		2	4			10
2	**Раздел 2. Работа с хранением, вводом, выводом данных**	**32**	**4**	**8**	**0**	**0**	**20**
2.1	Тема 1. Работа с файлами и исключениями		2	4			10
2.2	Тема 2. Работа с базами данных		2	4			10
3	**Раздел 3. Шаблоны и принципы программирования**	**41**	**6**	**10**	**0**	**0**	**25**

3.1	Тема 1. Паттерны проектирования		2	6			10
3.2	Тема 2. Тестирование и отладка в ООП		2	2			5
3.3	Тема 3. Принципы SOLID		2	2			10
	Итого	**144**	**18**	**36**	**0**	**0**	**90**

3.3 Содержание дисциплины

Раздел 1. Введение в объектно-ориентированное программирование

Определение объектно-ориентированной парадигмы. Основные понятия: классы, объекты, наследование. Преимущества и недостатки объектно-ориентированного программирования. Создание классов и объектов в Python. Методы и атрибуты классов. Конструкторы и деструкторы. Принцип наследования и его применение. Переопределение методов и атрибутов. Интерфейсы и абстрактные классы. Полиморфизм и его роль. Ограничение доступа к атрибутам и методам. Геттеры и сеттеры.

Раздел 2. Работа с хранением, вводом, выводом данных

Работа с файлами и потоками данных. Обработка исключений и ошибок. Механизмы обработки исключений в Python. Основы работы с базами данных. ORM (Object-Relational Mapping) и его применение. Создание, чтение, обновление и удаление данных.

Раздел 3. Шаблоны и принципы программирования

Основные понятия и принципы паттернов. Порождающие паттерны (Singleton, Factory Method). Структурные паттерны (Adapter, Decorator). Поведенческие паттерны (Observer, Strategy). Виды тестирования (юнит-тестирование, интеграционное тестирование). Отладка и инструменты для поиска ошибок и исключений. Принцип единственной ответственности (SRP). Принцип открытости/закрытости (OCP). Принцип подстановки Барбары Лисков (LSP). Принцип разделения интерфейса (ISP).

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1 Семинарские/практические занятия

Семинар 1-3. Создание простых классов и объектов, работа с методами и атрибутами
Семинар 4-6. Наследование и полиморфизм, создание иерархии классов.
Семинар 7-9. Работа с файлами, обмен данными
Семинар 10-12. Работа с базами данных, написание и тестирование запросов
Семинар 13-15. Паттерны проектирования, их применение в реальных сценариях
Семинар 16-18. Инкапсуляция, SOLID, применение принципов на практике

3.4.2 Лабораторные занятия

Не предусмотрены

3.5 Тематика курсовых проектов (курсовых работ)

Не предусмотрены

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

Не предусмотрены

4.2 Основная литература

1. Барков, И. А. Объектно-ориентированное программирование / И. А. Барков. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 700 с. — ISBN 978-5-507-47113-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/329549

2. Унгер, А. Ю. Объектно-ориентированное программирование : учебное пособие / А. Ю. Унгер. — Москва : РТУ МИРЭА, 2022. — 84 с. — ISBN 978-5-7339-1628-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/265691

3. Кувшинов, Д. Р. Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2023. — 104 с. — (Высшее образование). — ISBN 978-5-534-07559-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/533784

4.3 Дополнительная литература

1. Объектно-ориентированное программирование на C++ : учебник / И. В. Баранова, С. Н. Баранов, И. В. Баженова [и др.]. — Красноярск : СФУ, 2019. — 288 с. — ISBN 978-5-7638-4034-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/157572

2. Рик, Г. Объектно-Ориентированное Программирование / Г. Рик ; под редакцией Н. Комлева. — Москва : СОЛОН-Пресс, 2018. — 298 с. — ISBN 978-5-91359-285-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/107669

3. Чернышев, С. А. Основы программирования на Python : учебное пособие для среднего профессионального образования / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 349 с. — (Профессиональное образование). — ISBN 978-5-534-17056-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/532292

4.4 Электронные образовательные ресурсы

Не предусмотрены

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Anaconda
2. Python
3. Microsoft-Windows
4. Microsoft-Office

4.6 Современные профессиональные базы данных и информационные справочные системы

1. Научная электронная библиотека http://www.elibrary.ru
2. Российская государственная библиотека http://www.rsl.ru
3. Единое окно доступа к образовательным ресурсам Федеральный портал http://window.edu.ru
4. ЭБС «Университетская библиотека онлайн» https://biblioclub.ru/index.php

5 Материально-техническое обеспечение

1. Компьютерный класс с предустановленным программным обеспечением, указанным в п. 4.5, мультимедийное оборудование (проектор, персональный компьютер преподавателя).
2. Аудитория для лекционных, практических занятий. Оборудование и аппаратура: аудиторная доска, возможность использования мультимедийного комплекса.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

На первом занятии по дисциплине необходимо ознакомить студентов с порядком ее изучения (темами курса, формами занятий, текущего и промежуточного контроля), раскрыть место и роль дисциплины в системе наук, ее практическое значение, довести до студентов требования к форме отчетности и применения видов контроля.

При подготовке к практическим работам по перечню объявленных тем преподавателю необходимо уточнить план их проведения, продумать формулировки и содержание учебных вопросов, выносимых на обсуждение.

В ходе работы во вступительном слове раскрыть практическую значимость темы работы, определить порядок ее проведения, время на обсуждение каждого учебного вопроса. Применяя фронтальный опрос дать возможность выступить всем студентам, присутствующим на занятии.

В заключительной части работы следует подвести ее итоги: дать оценку выступлений каждого студента и учебной группы в целом. Раскрыть положительные стороны и недостатки проведенной работы. Ответить на вопросы студентов. Выдать задания для самостоятельной работы по подготовке к следующему занятию.

Методика преподавания дисциплины «Объектно-ориентированное программирование» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся:

– подготовка к выполнению и защита практических работ с помощью специализированного программного обеспечения;

– технологии анализа ситуаций для активного обучения, которые позволяют студентам соединить теорию и практику, представить примеры принимаемых решений и их последствий, демонстрировать различные позиции, формировать навыки оценки альтернативных вариантов в вероятностных условиях.

Обучение по дисциплине ведется с применением традиционных потоково-групповых информационно-телекоммуникационных технологий. При осуществлении образовательного

процесса по дисциплине используются следующие информационно-телекоммуникационные технологии: презентации с применением проектора и программы PowerPoint.

6.2 Методические указания для обучающихся по освоению дисциплины

Самостоятельная работа является одним из видов учебных занятий. Цель самостоятельной работы – практическое самостоятельное получение студентами навыков работы в программных продуктах, рассматриваемых в процессе изучения дисциплины.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Задачи самостоятельной работы студента:

- развитие навыков самостоятельной учебной работы;
- освоение содержания дисциплины;
- углубление содержания и осознание основных понятий дисциплины;
- использование материала, собранного и полученного в ходе самостоятельных занятий для эффективной подготовки к зачёту.

Виды внеаудиторной самостоятельной работы:

- самостоятельное изучение отдельных тем дисциплины;
- подготовка к практическим занятиям;
- оформление отчетов по выполненным практическим работам;

Для выполнения любого вида самостоятельной работы необходимо пройти следующие этапы:

- определение цели самостоятельной работы;
- конкретизация познавательной задачи;
- самооценка готовности к самостоятельной работе;
- выбор адекватного способа действия, ведущего к решению задачи;
- планирование работы (самостоятельной или с помощью преподавателя) над заданием;
- осуществление в процессе выполнения самостоятельной работы самоконтроля (промежуточного и конечного) результатов работы и корректировка выполнения работы;
- рефлексия;
- презентация работы.

7 Фонд оценочных средств

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций

- отчёты по практическим работам;
- контрольные работы;
- дифференцированный зачёт.

Оценочные средства текущего контроля успеваемости включают контрольные задания по практическим работам индивидуально для каждого обучающегося.

В результате освоения дисциплины (модуля) формируются следующие компетенции:

Код компетенции	Наименование компетенции выпускника
ОПК-5	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения

7.1 Методы контроля и оценивания результатов обучения

Перечень оценочных средств по дисциплине «Объектно-ориентированное программирование»

№ п/п	Вид контроля результатов обучения	Наименование контроля результатов обучения	Краткая характеристика контроля результатов обучения
1	Текущий	Практическая работа	Практическая работа выполняется индивидуально каждым студентом. По результатам выполнения работы студент оформляет отчёт, содержащий подробное описание проделанной работы с наглядным представлением результатов. Оформленный отчет студент сдает преподавателю на проверку в заранее установленный срок. При проверке преподаватель оценивает качество оформления, корректность проделанных шагов, результатов работы и выводов. Часть работ подразумевает устную защиту в формате доклада/презентации.
2	Текущий	Контрольная работа	Контрольная работа выполняется индивидуально каждым студентом. Задание подразумевает выполнения ряда шагов с использованием программного обеспечения, изучаемого в соответствующей теме. Результатом выполнения контрольной работы является электронный документ с кратким описанием выполненных шагов и полученных результатов. При проверке преподаватель оценивает как результат работы, так и пусть достижения результата.
3	Промежуточный	Дифференцированный зачёт	Промежуточная аттестация обучающихся в форме дифференцированного зачёта проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным

			планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно». Во время проведения дифференцированного зачёта студенту предлагается выполнить 3 практических задания. По результату выполнения студенту задаются вопросы на понимание выполненных задач. Количество дополнительных вопросов зависит от полноты ответа студента. К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Объектно-ориентированное программирование»

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю).

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
знать: Методы разработки алгоритмов в парадигме объектно-ориентированного программирования	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний методов разработки алгоритмов в парадигме объектно-ориентированного программирования	Обучающийся демонстрирует неполное соответствие следующих знаний методов разработки алгоритмов в парадигме объектно-ориентированного программирования. Допускаются значительные	Обучающийся демонстрирует частичное соответствие знаний методов разработки алгоритмов в парадигме объектно-ориентированного программирования. Допускаются незначительные	Обучающийся демонстрирует полное соответствие знаний методов разработки алгоритмов в парадигме объектно-ориентированного программирования. Свободно оперирует

		ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	ошибки, неточности, затруднения при аналитических операциях.	приобретенными знаниями.
уметь: Описывать задачу с точки зрения объектно-ориентированного подхода и подбирать соответствующие структуры для разработки алгоритма и программного кода	Обучающийся не умеет или в недостаточной степени умеет описывать задачу с точки зрения объектно-ориентированного подхода и подбирать соответствующие структуры для разработки алгоритма и программного кода	Обучающийся демонстрирует неполное соответствие следующих умений: описывать задачу с точки зрения объектно-ориентированного подхода и подбирать соответствующие структуры для разработки алгоритма и программного кода. Допускаются значительные ошибки, проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих умений: описывать задачу с точки зрения объектно-ориентированного подхода и подбирать соответствующие структуры для разработки алгоритма и программного кода Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся демонстрирует полное соответствие следующих умений: описывать задачу с точки зрения объектно-ориентированного подхода и подбирать соответствующие структуры для разработки алгоритма и программного кода. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
владеть: Навыками программирования на языке высокого уровня и алгоритмизации в объектно-ориентированной парадигме программирования	Обучающийся не владеет или в недостаточной степени владеет навыками программирования на языке высокого уровня и алгоритмизации в объектно-ориентированной парадигме программирования	Обучающийся в недостаточной степени владеет навыками программирования на языке высокого уровня и алгоритмизации в объектно-ориентированной парадигме программирования. Обучающийся испытывает значительные затруднения при применении	Обучающийся частично владеет навыками программирования на языке высокого уровня и алгоритмизации в объектно-ориентированной парадигме программирования. Навыки освоены, но допускаются незначительные ошибки, неточности, затруднения при	Обучающийся в полном объеме владеет навыками программирования на языке высокого уровня и алгоритмизации в объектно-ориентированной парадигме программирования. Свободно применяет полученные навыки в ситуациях

		навыков в новых ситуациях.	аналитических операциях, переносе умений на новые, нестандартные ситуации.	повышенной сложности.

Шкала оценивания промежуточной аттестации: дифференцированный зачёт

Шкала оценивания	Описание
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателям, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности, не испытывает затруднений при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует частичное соответствие знаний, умений, навыков приведенным в таблицах показателям, 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.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателям, 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.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует полное отсутствие или недостаточное соответствие знаний, умений, навыков приведенным в таблицах показателям, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент не может оперировать знаниями и умениями при их переносе на новые ситуации.

Шкала оценивания текущего контроля

Наименование контроля результатов обучения	Шкала оценивания	Описание
Практическая работа	Зачтено: набрано 2 и более баллов. Не зачтено: набрано 1 и менее баллов.	В качестве форм текущего контроля знаний студентов

	Критерии оценивания: Общий балл при оценке складывается из следующих показателей: - оформление работы соответствует требованиям – 1 балл; - все пункты задания выполнены в полном объёме – 2 балла; - изложение, описание и выводы по работе грамотны и полно описывают содержание практической работы – 2 балла.	используются отчеты по практической работе Отчёт по практической работе содержит описания ряда шагов по выполнению практической работы согласно заданию с подробным описанием проделанных действий и полученными результатами. Защита практических работ (если требуется) осуществляется индивидуально. Студентом предоставляется оформленный отчёт. Студенты, не выполнившие практическую работу, к защите не допускаются
Контрольная работа	Отлично - Работа высокого качества, уровень выполнения отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, либо некоторые из выполненных заданий содержат незначительные ошибки Хорошо - Уровень выполнения работы отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, некоторые виды заданий выполнены с ошибками. Удовлетворительно - Теоретическое содержание курса освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных программой заданий не выполнено; при дополнительной самостоятельной работе над материалом курса	Защита темы включает выполнения практического задания по изученному материалу в аудитории в течении одной пары. Билеты состоят из задач, позволяющих оценить сформированные компетенции. На решение отводится 1.5 часа.

	возможно повышение качества выполнения учебных заданий. Неудовлетворительно - Теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, предусмотренные программой задания не выполнены	

7.3 Оценочные средства

7.3.1 Текущий контроль

Практическая работа 1 - Создание простых классов и объектов:

Создайте класс "Студент", который содержит атрибуты для имени, возраста и средней оценки. Далее создайте несколько объектов этого класса и реализуйте методы для вывода информации о каждом студенте.

Практическая работа 2 - Наследование и полиморфизм:

Создайте базовый класс "Фигура" с методами для вычисления площади и периметра. От него унаследуйте классы "Прямоугольник" и "Круг". Реализуйте полиморфные методы вычисления площади и периметра для каждой фигуры.

Практическая работа 3 - Инкапсуляция, SOLID:

Создайте класс "Библиотека", который инкапсулирует список книг и методы для добавления, удаления и поиска книг. Примените принципы SOLID, разделив функциональность на отдельные классы, например, "Каталог" и "МенеджерКниг".

Практическая работа 4 - Работа с файлами и исключениями:

Напишите программу, которая считывает данные из файла, проводит какие-то операции (например, вычисления) и записывает результат в другой файл. Обработайте возможные исключения, такие как отсутствие файла или ошибки в данных.

Практическая работа 5 - Паттерны проектирования:

Выберите один паттерн проектирования (например, "Одиночка" или "Фабрика") и продемонстрируйте его применение в реальном сценарии. Напишите код, в котором используются принципы выбранного паттерна для улучшения структуры приложения.

Практическая работа 6 - Работа с базами данных:

Создайте базу данных студентов, используя SQL. Напишите программу, которая позволяет добавлять, изменять и удалять записи о студентах, а также выполнять запросы для получения информации, например, список студентов с определенной средней оценкой.

Контрольная работа 1 - Применение ООП:

Реализуйте систему банковских счетов с использованием ООП. Создайте классы для клиентов, банковских счетов, операций по счетам. Продемонстрируйте принципы наследования, полиморфизма и инкапсуляции.

Контрольная работа 2 - Принципы SOLID:

Рассмотрите реальную задачу (например, систему заказов) и покажите, как вы можете применить каждый из принципов SOLID (Принцип единственной ответственности,

Открытость/закрытость, Принцип подстановки Барбары Лисков и т.д.) для создания гибкой и расширяемой системы.

Контрольная работа 3 - Работа с данными и базами:

Разработайте веб-приложение, которое позволяет пользователям регистрироваться, авторизовываться и добавлять комментарии к статьям. Используйте базу данных для хранения информации о пользователях, статьях и комментариях. Реализуйте функциональность обмена данными между клиентом и сервером.

7.3.2 Промежуточная аттестация

Что такое объектно-ориентированная парадигма?	ОПК-5
Какие основные понятия включает объектно-ориентированное программирование?	ОПК-5
Каковы преимущества и недостатки объектно-ориентированного программирования?	ОПК-5
Как создать класс и объект в Python?	ОПК-5
Что такое методы и атрибуты классов?	ОПК-5
Какие методы выполняются автоматически при создании и удалении объекта?	ОПК-5
Представьте практический пример использования классов и объектов.	ОПК-5
Как применяется принцип наследования в объектно-ориентированном программировании?	ОПК-5
Что такое переопределение методов и атрибутов?	ОПК-5
Как создать абстрактный класс и интерфейс?	ОПК-5
Что такое полиморфизм и как он реализуется?	ОПК-5
Какие механизмы обеспечивают инкапсуляцию в языке программирования?	ОПК-5
Что такое геттеры и сеттеры, и зачем они используются?	ОПК-5
Приведите примеры реализации инкапсуляции.	ОПК-5
Какие механизмы обработки ошибок и исключений существуют?	ОПК-5
Объясните принцип единственной ответственности (SRP).	ОПК-5
Как принцип открытости/закрытости (OCP) влияет на расширяемость кода?	ОПК-5
Что означает принцип подстановки Барбары Лисков (LSP)?	ОПК-5
Какой смысл принципа разделения интерфейса (ISP)?	ОПК-5
В чем заключается принцип инверсии зависимостей (DIP)?	ОПК-5
Какие методы позволяют работать с файлами и потоками данных?	ОПК-5
Какие основные исключения могут возникнуть при работе с файлами?	ОПК-5
Как обрабатываются исключения в Python?	ОПК-5
Что такое паттерны проектирования и для чего они используются?	ОПК-5
Какой паттерн проектирования реализует создание единственного экземпляра класса?	ОПК-5
Что делает паттерн "Фабричный метод"?	ОПК-5
Как работает паттерн "Адаптер" и в каких ситуациях он используется?	ОПК-5
Что такое паттерн "Декоратор" и как он расширяет функциональность объекта?	ОПК-5
Как паттерн "Наблюдатель" позволяет реализовать обратную связь между объектами?	ОПК-5
Какой паттерн позволяет выбирать алгоритм выполнения во время выполнения программы?	ОПК-5
Что такое ORM и какие преимущества он предоставляет?	ОПК-5
Какие основные операции можно выполнять с данными в базе данных?	ОПК-5
Каким образом происходит создание и подключение к базе данных в Python?	ОПК-5

Приведите пример работы с базой данных с использованием языка SQL.	ОПК-5
В чем заключается юнит-тестирование?	ОПК-5
Какие принципы написания хороших тестов соблюдаются при тестировании классов и методов?	ОПК-5
Какие виды ошибок можно обнаружить с помощью инструментов отладки?	ОПК-5
Какие инструменты используются для поиска ошибок в программном коде?	ОПК-5