

Семестр (<Курс>.<Семестр на курсе>)	1 (1.1)		Итого	
Недель	17 1/6			
Вид занятий	уп	рп	уп	рп
Лекции	4	4	4	4
Практические	6	6	6	6
Итого ауд.	10	10	10	10
Контактная работа	10	10	10	10
Сам. работа	24	24	24	24
Часы на контроль	2	2	2	2
Итого	36	36	36	36

Программу составил(и):

Кулемин Александр Николаевич

Рабочая программа дисциплины

Основы программирования

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - магистратура по направлению подготовки 38.04.05 Бизнес-информатика (приказ Минобрнауки России от 12.08.2020 г. № 990)

составлена на основании учебного плана:

38.04.05 Бизнес-информатика

Одобрена Ученым советом вуза

Протокол от 24.11.2025 № 2.

Одобрена учебно-методической комиссией

Протокол 17.11.2025 № 1

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
1.1	Цель освоения дисциплины:
1.2	Формирование у обучающихся базовых знаний в области алгоритмизации и программирования, понимания принципов работы программных систем и навыков практического использования языка Python для решения типовых задач бизнес-информатики.
1.3	Задачи изучения дисциплины:
1.4	- Ознакомить с основными понятиями программирования и структурой программ;
1.5	- Научить использовать базовые конструкции Python (переменные, условия, циклы, функции);
1.6	- Сформировать навыки анализа и обработки данных средствами Python;
1.7	- Показать возможности автоматизации вычислений и типовых задач бизнес-анализа.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	ФТД
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Обучающийся должен владеть основами информатики, логики и ПК на уровне пользователя.
2.2	Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Подготовка к процедуре защиты и защита выпускной квалификационной работы
2.2.2	Производственная практика (проектно-технологическая) (часть 2)

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ИНДИКАТОРЫ ИХ ДОСТИЖЕНИЯ	
ОПК-2: Способен учитывать конкретные условия выполняемых задач и разрабатывать инновационные решения при управлении проектами и процессами в сфере информационно-коммуникационных технологий	
ОПК-2.И-2: Понимает возможности и направления использования современных информационных и цифровых технологий для поддержки деятельности организации	
Знать: Принципы структурного и модульного программирования, архитектурные шаблоны (функциональная декомпозиция, MVC, модульность) Уметь: Применять методы структурного анализа для построения архитектуры простых программных решений Владеть: Навыками описания архитектуры и структуры программы с использованием UML-диаграмм и псевдокода	

ПК-1: Управление ИТ- сервисами (Способен управлять непрерывностью бизнеса с использованием ИТ)	
ПК-1.И-3: Выявляет потребности заинтересованных сторон в новых ИТ-сервисах, осуществляет их проектирование, разработку и внедрение	
Знать: Основы проектирования прикладных ИТ-сервисов, методы сбора и обработки данных, принципы верификации и тестирования Уметь: Разрабатывать программные решения на Python для решения типовых бизнес-задач; применять тестирование и документирование Владеть: Средствами Python, инструментами контроля версий, форматами документации	

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Примечание
	Раздел 1.			
1.1	Тема 1. Программная инженерия: понятия, жизненный цикл, принципы проектирования ПО /Лек/	1	2	
1.2	Тема 1. Программная инженерия: понятия, жизненный цикл, принципы проектирования ПО /Пр/	1	2	
1.3	Тема 1. Программная инженерия: понятия, жизненный цикл, принципы проектирования ПО /Ср/	1	4	
1.4	Тема 2. Основы алгоритмизации и структурного программирования на Python, работа с данными /Лек/	1	1	

1.5	Тема 2. Основы алгоритмизации и структурного программирования на Python, работа с данными /Пр/	1	2	
1.6	Тема 2. Основы алгоритмизации и структурного программирования на Python, работа с данными /Ср/	1	8	
1.7	Тема 3. Проектирование и реализация модулей и функций /Лек/	1	1	
1.8	Тема 3. Проектирование и реализация модулей и функций /Пр/	1	2	
1.9	Тема 3. Проектирование и реализация модулей и функций /Ср/	1	12	
1.10	/Зачёт/	1	2	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации

Темы для обсуждения на практических занятиях:

Тема 1. Жизненный цикл и принципы проектирования программного обеспечения

- Жизненный цикл ПО (от требований до сопровождения), роль модульности и инкапсуляции.
- Принципы проектирования: «разделяй и властвуй», повторное использование, читаемость кода.

Качество кода и базовые практики

- Именованное, комментарий, стиль (PEP 8 как ориентир), простые шаблоны (функции-утилиты).
- Верификация и отладка: утверждения (assert), простые модульные проверки.

Тема 2. Алгоритмизация и управление потоком исполнения

- Типовые конструкции: последовательность, ветвление, цикл; следование, выбор, повторение.
- Трассировка выполнения, пошаговая отладка, чтение трасс-сообщений. (

Работа с данными в Python

- Базовые типы и структуры: число, строка, список, словарь, множество, кортеж.
- Ввод/вывод и форматирование, чтение/запись файлов (UTF-8), простая обработка CSV.)

Тема 3. Проектирование функций

- Подпись функции, параметры и значения по умолчанию, область видимости, возвращаемые значения.
- Чистые функции, побочные эффекты, тестируемость.

Модули и структура проекта

- Импорт модулей, организация пакета (init.py), точка входа if __name__ == "__main__": (ПК-1.И-3)
- Простая документация (docstring), структура папок, требования к README.

Примеры кейсовых заданий (к теме 3):

Кейс 1. «Утилита расчёта показателей»

Ситуация. Необходимо из текстового файла с продажами (CSV) посчитать средний чек и медиану.

Данные/ограничения. Файл sales.csv со столбцами: date;customer;amount. Ошибочные строки следует пропускать с логированием.

Задание. 1) Спроектируйте модуль stats.py с функциями read_sales(path), avg(values), median(values); 2) Напишите точку входа main.py, принимающую путь к файлу; 3) Добавьте мини-проверки (assert) и docstring к функциям.

Кейс 2. «Конвертер форматов»

Ситуация. Нужно преобразовать файл с парами ключ-значение из формата key=value в JSON.

Данные/ограничения. Вход может содержать пустые строки и комментарии, кодировка UTF-8.

Задание. 1) Спроектируйте функции parse_kv(text) и to_json(mapping, path) в модуле io_utils.py; 2) Реализуйте обработку некорректных строк; 3) Напишите сценарий командной строки с аргументами --in и --out.

Кейс 3. «Мини-библиотека строковых утилит»

Ситуация. Команде нужна небольшая библиотека для работы со строками (нормализация пробелов, подсчёт слов, реверс).

Данные/ограничения. Функции должны быть чистыми, покрыты простыми проверками.

Задание. 1) Спроектируйте text_utils.py с функциями normalize_ws, word_count, reverse_words; 2) Добавьте docstring с примерами; 3) Соберите пакет utils/ с init.py и примером использования.

5.2. Темы письменных работ

не предусмотрены

5.3. Фонд оценочных средств

Тестовые задания:

Вопросы с одним правильным ответом:

1. Что описывает «жизненный цикл ПО»?

- Только программирование
- Этапы от требований и проектирования до тестирования и сопровождения
- Лишь развертывание
- Только поддержку

2. Какая цель модульности?

- Увеличить длину файлов
- Упростить сопровождение и повторное использование
- Запретить функции
- Исключить документацию

3. Что делает оператор if?
- a) Повторяет блок
 - b) Выбирает ветвь исполнения по условию
 - c) Завершает программу
 - d) Объявляет функцию
4. Какой тип данных изменяемый?
- a) Кортеж
 - b) Строка
 - c) Список
 - d) Число
5. Как передать аргументы скрипту?
- a) Через переменную this
 - b) Через sys.argv/парсер аргументов
 - c) Никак
 - d) Через заголовок файла
6. Что возвращает функция без оператора return?
- a) 0
 - b) Пустую строку
 - c) None
 - d) Последний параметр
7. Где видны локальные переменные функции?
- a) Во всей программе
 - b) Только внутри функции
 - c) В импортирующем модуле
 - d) В файловой системе
8. Что делает выражение list.append(x)?
- a) Создает копию списка
 - b) Добавляет элемент в конец списка
 - c) Удаляет элемент
 - d) Сортирует список
9. Как открыть текстовый файл для чтения в UTF-8?
- a) open(path)
 - b) open(path, 'r', encoding='utf-8')
 - c) read(path, utf8=True)
 - d) file(path)
10. Для чего используются docstring?
- a) Для комментариев к коммитам
 - b) Для описания назначения функции/модуля и примеров
 - c) Для ускорения выполнения
 - d) Для подсветки синтаксиса
11. Что означает «чистая функция»?
- a) Не имеет параметров
 - b) Не зависит от внешнего состояния и не меняет его
 - c) Использует глобальные переменные
 - d) Печатает результат на экран
12. Какая конструкция организует повторение?
- a) if
 - b) for/while
 - c) def
 - d) return
13. Как импортировать функцию f из модуля tools?
- a) use tools.f
 - b) import tools.f()
 - c) from tools import f
 - d) include tools
14. Что делает assert condition?
- a) Всегда возвращает True
 - b) Прерывает выполнение при ложном условии
 - c) Запускает тесты
 - d) Форматирует строку
15. Как создать словарь?
- a) {} или dict()
 - b) []
 - c) ()
 - d) <>
- 16 (один ответ). Где размещается точка входа консольного сценария?
- a) В случайном месте
 - b) В блоке if __name__ == "__main__":

- c) В docstring
- d) В комментарии
- 17. Что делает return?
- a) Завершает функцию и возвращает значение
- b) Печатает значение
- c) Завершает программу
- d) Импортирует модуль
- 18. Что такое «повторное использование кода»?
- a) Копирование файла
- b) Проектирование общих функций/модулей для разных задач
- c) Дублирование строк
- d) Переименование переменных
- 19. Чем список отличается от кортежа?
- a) Список неизменяемый, кортеж изменяемый
- b) Оба неизменяемы
- c) Список изменяемый, кортеж неизменяемый
- d) Оба изменяемы
- 20. Как объединить строки с разделителем ,?
- a) ','.split(list)
- b) ','.join(list_of_strings)
- c) concat(',', list)
- d) + ',' +
- 21. Зачем нужны параметры по умолчанию?
- a) Чтобы функция всегда падала
- b) Чтобы упростить вызов и задать стандартное поведение
- c) Чтобы скрыть ошибки
- d) Чтобы усложнить интерфейс

Вопросы с множественным выбором (выберите три правильных варианта ответа):

- 22. Признаки хорошей функции:
- a) Выполняет одну чёткую задачу
- b) Имеет понятные имя и параметры
- c) Скрывает логику в глобальных переменных
- d) Имеет примеры в docstring
- 23. Что помогает читать код?
- a) Говорящие имена переменных
- b) Лишние комментарии без смысла
- c) Короткие функции
- d) Структура папок и модулей
- 24. Что относится к структуре проекта?
- a) Папка utils/ с __init__.py
- b) Единый стиль именования
- c) Отсутствие README
- d) Точка входа
- 25. Что относится к обработке ошибок при чтении файлов? (выберите два правильных варианта ответа)
- a) Игнорировать исключения
- b) Сообщать об ошибочных строках
- c) Прерывать программу без сообщения
- d) Пропускать некорректные записи с учётом требований.

5.4. Перечень видов оценочных средств и критерии оценивания

Контроль результатов освоения дисциплины является формой управления качества образования.

Контроль результатов освоения дисциплины осуществляется в виде текущего контроля и промежуточной аттестации (зачета).

Система текущего контроля включает:

1. контроль работы обучающихся на лекционных и практических занятиях;
2. контроль участия в совместной работе группы;
3. контроль выполнения индивидуальных заданий;
4. контроль выполнения обучающимися заданий для самостоятельной работы.

Промежуточная аттестация осуществляется в виде зачета, который проводится в устной либо письменной форме.

Задачи промежуточной аттестации:

1. определить общий уровень освоения дисциплины в целом;
2. определить уровень основных знаний по каждой теме курса;
3. объективизировать результат контроля, минимизировать возможную субъективность преподавателя.

Рубрики и критерии оценивания

Элемент контроля/Критерий/Индикаторы/Уровни (0–5)/Комментарий к уровню «5»

Семинар/Понимание конструкций и структур, участие/ПК-1.И-3, ОПК-2.И-2/0/3/5/Точные объяснения, корректные

примеры, увязка с задачами организации

Кейс 1 (stats)/Проектирование функций и корректность расчётов/ПК-1.И-3, ОПК-2.И-2/0/3/5/Корректные формулы, обработка ошибок, интерпретация результатов для бизнеса

Кейс 2 (конвертер)/Проектирование модулей и ввод/вывод/ПК-1.И-3, ОПК-2.И-2/0/3/5/Структура модулей и сценария, понимание практического применения утилиты

Кейс 3 (строковые утилиты)/Чистые функции и документация/ПК-1.И-3, ОПК-2.И-2/0/3/5/Чистые функции с docstring-примерами, акцент на удобстве использования

Тест/Корректность ответов/ПК-1.И-3, ОПК-2.И-2/%→балл/≥90% — 5; 75–89% — 4; 60–74% — 3; <60% — 2

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

Основная литература

Лаврищева, Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. — Москва : Издательство Юрайт, 2025. — 280 с. — (Высшее образование). — ISBN 978-5-534-01056-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561899> (дата обращения: 17.11.2025).

Программирование на высоком уровне Python: учебное пособие. — М.: Юрайт, 2025. — 304 с. URL: <https://urait.ru/bcode/437489> (дата обращения: 17.11.2025).

Федоров Д. Ю. Программирование на python: учебное пособие для вузов. 6-е изд., перераб. и доп. — М.: Юрайт, 2025. — 187 с. URL: <https://urait.ru/bcode/556864> (дата обращения: 17.11.2025).

Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 146 с. — (Высшее образование). — ISBN 978-5-534-18197-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562413> (дата обращения: 17.11.2025).

Дополнительная литература

Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебник для вузов / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 91 с. — (Высшее образование). — ISBN 978-5-534-21288-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563580> (дата обращения: 17.11.2025).

Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567821> (дата обращения: 17.11.2025).

Судоплатов, С. В. Математика: математическая логика и теория алгоритмов : учебник и практикум для среднего профессионального образования / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2025. — 255 с. — (Профессиональное образование). — ISBN 978-5-534-10930-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566184> (дата обращения: 17.11.2025).

6.2. Электронные учебные издания и электронные образовательные ресурсы

Э1	Сетевые ресурсы Государственного автономного учреждения культуры Свердловской области «Свердловская областная универсальная научная библиотека им. В.Г. Белинского». Режим доступа: https://book.uraic.ru/el_library/resursy_dostup
Э2	Образовательная платформа Юрайт. Режим доступа: https://urait.ru/
Э3	Научная библиотека Cyberleninka. Режим доступа: https://cyberleninka.ru/

6.3.1 Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства

6.3.1.1	Операционная система Astra Linux
6.3.1.2	Офисная система LibreOffice
6.3.1.3	Python 3.x
6.3.1.4	Контур.Класс
6.3.1.5	Контур.Толк
6.3.1.6	1С:Университет ПРОФ
6.3.1.7	Среда электронного обучения 3LK (русский Moodle)

7. МТО (оборудование и технические средства обучения)

7.1	Учебные аудитории для проведения учебных занятий, предусмотренных программой магистратуры, оснащенные оборудованием и техническими средствами обучения:	
	Адрес (местоположение) учебных аудиторий	Перечень оборудования и технических средств обучения
	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 3 площадью 66,2 м² в помещении 501	Стол – 13 шт. Стул – 25 шт. ПК (с возможностью подключения к сети «Интернет») – 1 шт. Телевизор (75”) – 1 шт. Беспроводная система презентаций – 1 шт. Спикерфон – 1 шт. ПК (ноутбук) – 12 шт.
	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 19 площадью 31,7 м² в помещении 501	Стол – 7 шт. Стул – 13 шт. ПК (с возможностью подключения к сети «Интернет») – 1 шт. Магнитно-маркерная доска – 1 шт. Телевизор (75”) – 1 шт.
7.2	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 21 площадью 35,6 м² в помещении 501	Стол – 7 шт. Стул – 13 шт. ПК (с возможностью подключения к сети «Интернет») – 1 шт. Магнитно-маркерная доска – 1 шт. Телевизор (75”) – 1 шт.
	Помещение для самостоятельной работы обучающихся, оснащенное компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде университета:	
	Адрес (местоположение) помещения для самостоятельной работы обучающихся	Перечень компьютерной техники и оборудования
	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 5 площадью 44,3 м² в помещении 501	ПК (ноутбук) – 9 шт. Стол – 11 шт. Стул – 11 шт.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для успешного освоения дисциплины рекомендуется ознакомиться с рабочей программой, целями и задачами дисциплины, её связями с другими дисциплинами образовательной программы. Также рекомендуется изучить основную и дополнительную литературу, перечень заданий. Перед лекцией просматривать конспект предыдущей лекции для более глубокого восприятия материала. При подготовке к практическим занятиям изучить основную литературу, ознакомиться с дополнительной и методической литературой, учесть рекомендации преподавателя. Важно также опираться на конспекты лекций. В ходе занятий вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных понятий, явлений и процессов, научные выводы и практические рекомендации по их применению. При пропуске занятия самостоятельно изучить материал и предоставить преподавателю отчёт по пропущенной теме в часы индивидуальных консультаций. При подготовке к текущей аттестации тщательно изучать материал по блокам тем, акцентировать внимание на определениях, терминах, содержании понятий. Для этого рекомендуется использовать основную литературу, а также конспекты лекций, составленные в ходе изучения всей дисциплины. При подготовке к промежуточной аттестации в дополнение к изучению конспектов лекций, словарей, учебных пособий, целесообразно пользоваться учебной литературой, рекомендованной в программе, и электронными ресурсами. При подготовке к зачёту или экзамену организовать учёбу так, чтобы все виды работ и заданий, предусмотренные рабочей программой, были выполнены в срок. Основное в подготовке — это повторение всего материала учебной дисциплины.