

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Б1.В.03
(индекс дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Алгоритмы и программирование на основе Python

(наименование дисциплины)

по направлению подготовки

09.03.04 Программная инженерия

направленность (профиль)

Программная инженерия с применением ИИ - технологий

Форма обучения: заочная

Год набора: 2024

Общая трудоемкость: 4 ЗЕ

Распределение часов дисциплины по семестрам

Семестр	3	Итого
Форма контроля	Экзамен	
Вид занятий		
Лекции	4	4
Лабораторные		
Практические		
Руководство: курсовые работы (проекты) / РГР		
Промежуточная аттестация	0,35	0,35
Контактная работа	4,35	4,35
Самостоятельная работа	131	131
Контроль	8,65	8,65
Итого	144	144

Рабочую программу составил(и):

Доцент кафедры института цифровых технологий, канд.тех.наук, Хрипунов Н.В.

(должность, ученое звание, степень, Фамилия И.О.)

(должность, ученое звание, степень, Фамилия И.О.)

Рецензирование рабочей программы дисциплины:



Отсутствует



Рабочая программа дисциплины составлена на основании ФГОС ВО и учебного плана направления подготовки

09.03.04 Программная инженерия

Срок действия рабочей программы дисциплины до «31» августа 2031 г.

УТВЕРЖДЕНО

На заседании института цифровых технологий

(протокол заседания № 1 от «05» сентября 2025 г.).

1. Цель освоения дисциплины

Цель освоения дисциплины – сформировать у обучающихся знания и практические навыки по алгоритмизации и программированию на языке Python.

2. Место дисциплины в структуре ОПОП ВО

Дисциплины и практики, на освоении которых базируется данная дисциплина: основы программирования.

Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее: объектно-ориентированное программирование.

3. Планируемые результаты обучения

Формируемые и контролируемые компетенции (код и наименование)	Индикаторы достижения компетенций (код и наименование)	Планируемые результаты обучения
ПК-1 Способен формализовать и алгоритмизировать задачи для разработки программного обеспечения	ПК-1.1. Знает понятие алгоритма, методы формализации и способы описания алгоритмов	Знать: основные структуры алгоритмов (следование, ветвление, цикл); методы формализации (диаграммы потоков данных, Use Case диаграммы); нотации для описания (псевдокод, блок-схемы). Уметь: декомпозировать задачу на подзадачи. Владеть: навыками создания технического задания на основе формализованных требований.
	ПК-1.2. Умеет описать и построить алгоритм задачи для разработки программного обеспечения	Знать: критерии оценки качества алгоритма (сложность, эффективность). Уметь: выбирать и описывать оптимальный алгоритм для решения задачи. Владеть: навыками моделирования бизнес-процессов и их перевода в алгоритмическую форму.
	ПК-1.3. Владеет навыками алгоритмизации, проектирования и разработки программного обеспечения	Знать: принципы нисходящего и восходящего проектирования. Уметь: переходить от алгоритма к архитектуре и коду.

Формируемые и контролируемые компетенции (код и наименование)	Индикаторы достижения компетенций (код и наименование)	Планируемые результаты обучения
		Владеть: навыками использования UML для проектирования и прототипирования.
ПК-2. Способен разрабатывать и отлаживать программный код на языках программирования	ПК-2.1. Знает основные правила конструирования и отладки программного кода	Знать: принципы чистого кода, стили кодирования; методы отладки (пошаговое выполнение, логирование). Уметь: читать и анализировать чужой код. Владеть: навыками использования отладчика в IDE.
	ПК-2.2. Умеет отлаживать программный код на языках программирования	Знать: типовые ошибки и исключения в выбранном языке программирования. Уметь: идентифицировать и локализовывать источник ошибки; использовать профилировщик для поиска "узких мест". Владеть: навыками написания модульных тестов для предотвращения регрессий.
	ПК-2.3. Владеет навыками разработки и отладки программного кода на языках программирования	Знать: инструменты непрерывной интеграции (CI) для автоматической сборки и тестирования. Уметь: писать надежный, эффективный и поддерживаемый код. Владеть: практическими навыками разработки на одном из языков включая отладку и рефакторинг.

4. Структура и содержание дисциплины

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
Модуль 1. Основы языка Python	Лек1	Процесс создания проекта в Python. Типы данных и операции языка Python	3	2			
	Ср	Линейный алгоритм	3	8			
	Ср	Программирование алгоритмов линейной структуры	3	8	8		Отчёт по практическому заданию 1
	Ср	Разветвляющийся алгоритм	3	7			
	Ср	Программирование алгоритмов разветвляющейся структуры	3	8	8		Отчёт по практическому заданию 2
	Ср	Циклический алгоритм	3	7			
	Ср	Программирование алгоритмов циклической структуры	3	8	8		Отчёт по практическому заданию 3
	Ср	Работа с кортежами, списками и словарями	3	8			
	Ср	Программирование алгоритмов формирования и обработки списков	3	8	8		Отчёт по практическому заданию 4
	Ср	Работа со строками	3	7			
	Ср	Обработка строковых данных	3	8	8		Отчёт по практическому заданию 5
	Ср	Функции, модули в языке Python	3	8			
	Ср	Работа с функциями	3	8	8		Отчёт по практическому заданию 6

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
	Ср	Работа с файлами	3	7			
	Ср	Работа с текстовыми файлами	3	8	10		Отчёт по практическому заданию 7
Модуль 2. Объектно-ориентированное программирование	Лек2	Основные методы и предварительная обработка данных.	3	2	-		
	Ср	Классы в языке Python. Методы класса	3	8	-		
	Ср	Наследование и инкапсуляция	3	7	-		
	Ср	Объектно-ориентированное программирование	3	8	10		Отчёт по практическому заданию 8
	Контроль	Экзамен	3	8,65	40		Итоговый тест
Итого:				144	100		

Схема расчета итогового балла: Текущий рейтинг (все занятия и промежуточные тесты) + результат итогового теста

5. Образовательные технологии

В процессе изучения дисциплины используется технология традиционного обучения (лекции, практические работы, самостоятельная работа обучающегося)

6. Методические указания по освоению дисциплины

Для успешного освоения дисциплины необходимы посещение обучающимися лекционных и практических занятий, самостоятельная работа обучающихся с лекционным материалом и учебной литературой.

Изучение дисциплины требует систематического и последовательного накопления знаний, следовательно, пропуски отдельных тем не позволяют глубоко освоить предмет.

В ходе лекционных занятий полезно задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

Обучающийся может дополнить список предложенной литературы современными источниками, не представленными в списке, и в дальнейшем использовать собственные подготовленные учебные материалы при написании курсовых и выпускных квалификационных работ.

Обучающимся следует

- при подготовке к практическим занятиям обязательно использовать не только лекции, учебную литературу, но и другие источники;
- в начале занятий задавать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и использовании при решении задач, предложенных для самостоятельного решения;
- на занятиях доводить каждую задачу до окончательного ответа, демонстрировать понимание проведенных расчетов (рассуждений), в случае затруднений обращаться к преподавателю.

Для того чтобы практические занятия приносили максимальную пользу, необходимо помнить, что решение задач проводится по рассмотренному на лекциях материалу и связано, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться обучающимся на практических занятиях как в результате обсуждения и анализа лекционного материала, так и в процессе решения задач. При этих условиях обучающийся не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (что очень важно) для активной проработки лекционного материала.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если обучающийся видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений (рассуждений, преобразований) составить краткий план решения проблемы (задачи). Решение задач следует излагать подробно, вычисления (рассуждения, преобразования) располагать в строгом порядке. Решение при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Полезно (если это возможно) решать задачу несколькими способами и сравнивать полученные результаты. Решение задач определённого типа нужно продолжать до приобретения твердых навыков в их решении.

Самостоятельная работа обучающихся по предмету организуется в следующих формах:

- 1) самостоятельное изучение основного теоретического материала, ознакомление с дополнительной литературой, Интернет-ресурсами;
- 2) решение профессиональных задач из реальной предметной области.

В качестве учебно-методического обеспечения самостоятельной работы используется основная и дополнительная литература по предмету, Интернет-ресурсы, материал лекций, указания, выданные преподавателем при проведении практических работ.

Подготовка к экзамену способствует закреплению, углублению и систематизации знаний, получаемых в процессе обучения. Готовясь к экзамену, обучающийся ликвидирует имеющиеся пробелы в знаниях, упорядочивает свои знания. На зачете обучающийся демонстрирует как теоретические знания, приобретённые в процессе обучения по данной учебной дисциплине, так и навыки их практического использования при решении задач.

Необходимо ориентировать обучающихся на систематическую подготовку к занятиям в течение семестра, поскольку это позволит освоить основы изучаемой дисциплины, а время экзаменационной сессии можно будет использовать для систематизации уже имеющихся знаний.

7. Оценочные средства

7.1. Паспорт оценочных средств

Семестр	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
3	ПК-1, ПК-2	Тестовые задания Вопросы к экзамену 1-80 Практические работы 1-8

7.2. Типовые задания или иные материалы, необходимые для текущего контроля

7.2.1. Отчеты по практическим работам

(наименование оценочного средства)

Типовые примеры заданий

Практическая работа 1. Программирование алгоритмов линейной структуры.

Цель работы. Освоить построение и реализацию алгоритмов линейной структуры на языке Python; закрепить навыки работы с вводом данных, базовыми арифметическими операциями, строковыми преобразованиями и последовательным выполнением команд.

Задание 1. Простейшие вычисления

Написать программу, которая запрашивает у пользователя два числа и выводит их сумму, разность, произведение и частное. Предусмотреть проверку деления на ноль.

Задание 2. Вычисление значения формулы

Реализовать программу, вычисляющую выражение:

$$f(x, y) = \frac{x^2 + y^2}{x - y}$$

Значения x и y вводятся пользователем. Учесть случай, когда знаменатель равен нулю.

Задание 3. Перевод единиц измерения

Написать программу, которая получает расстояние в метрах и выводит это расстояние в километрах, сантиметрах и миллиметрах.

Задание 4. Работа со строками

Создать программу, которая запрашивает строку у пользователя и выводит:

- длину строки,
- первый символ,
- последний символ, строку в верхнем регистре.

Задание 5. Повышенный уровень

Пользователь вводит трёхзначное число. Программа должна вывести сумму его цифр и число, записанное в обратном порядке.

Задание 6. Форматированный вывод

Написать программу, которая получает имя, фамилию и возраст пользователя и выводит фразу:

«Здравствуйте, <Имя> <Фамилия>. Вам <Возраст> лет.»

Вопросы к защите

1. Что называется алгоритмом линейной структуры?
2. Как выполняются команды в линейном алгоритме?
3. Какие числовые типы данных используются в Python?
4. Как преобразовать строку, введённую через input(), в число?
5. Какие арифметические операции доступны в Python?
6. Какие ошибки могут возникать при вычислениях?
7. Что означает деление на ноль и как его избежать?
8. Что такое строка в Python?
9. Как получить символ строки по индексу?
10. Какие способы форматирования текста существуют в Python?

Требования к оформлению

Отчет должен содержать подробное описание (включая иллюстративный материал) последовательности действий, проделанных обучающимся для выполнения заданий.

Процедура оценивания

Оценка выполненной работы проводится по критериям:

1. Наличие всей существенной информации по работе
2. Точность и полнота предоставляемых сведений
3. Непротиворечивость приводимой информации
4. Правильность интерпретаций и выводов, которые сделаны по результатам работы
5. Степень достижения обучающимся поставленной цели
6. Обоснованность применяемого решения
7. Грамотность (содержательная) используемых формулировок

Критерии оценки:

Критерии оценки за отчеты по практическим работам 1-6:

- Работа выполнена в полном объеме с соблюдением необходимой последовательности действий; аккуратно, четко и без ошибок выполнил отчет, вывод исчерпывающий и доказательный. При защите отчета ответил на все вопросы по теме; хорошо ориентируется в материале, умеет определить взаимосвязь факторов и их влияние на конечную цель, умеет графически отобразить важнейшие функциональные зависимости – 8 баллов
- Работа выполнена в полном объеме с соблюдением необходимой последовательности действий; обучающийся без ошибок выполнил отчет, вывод исчерпывающий. При защите отчета хорошо разбирается в материале, но не уверен и неполно отвечает на вопросы. Способность к обобщению причинно-следственных связей важнейших факторов выражена недостаточно – 7-6 баллов;
- Работа выполнена не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы; выполнен с несущественными

замечаниями. Вывод по работе не раскрывает сути работы. Владение понятийным аппаратом темы недостаточны – 5-3 балла;

- Обучающийся выполнил работу не полностью или объем выполненной части работы не позволяет сделать правильных выводов. В ответах на вопросы есть грубые ошибки. Нет знания принципиальных теоретических положений темы –1-2 балла

Критерии оценки за отчеты по практическим работам 7-8:

- Работа выполнена в полном объеме с соблюдением необходимой последовательности действий; аккуратно, четко и без ошибок выполнил отчет, вывод исчерпывающий и доказательный. При защите отчета ответил на все вопросы по теме; хорошо ориентируется в материале, умеет определить взаимосвязь факторов и их влияние на конечную цель, умеет графически отобразить важнейшие функциональные зависимости – 10 баллов

- Работа выполнена в полном объеме с соблюдением необходимой последовательности действий; обучающийся без ошибок выполнил отчет, вывод исчерпывающий. При защите отчета хорошо разбирается в материале, но не уверен и неполно отвечает на вопросы. Способность к обобщению причинно-следственных связей важнейших факторов выражена недостаточно – 7-9 баллов;

- Работа выполнена не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы; выполнен с несущественными замечаниями. Вывод по работе не раскрывает сути работы. Владение понятийным аппаратом темы недостаточны – 4-6 баллов;

- Обучающийся выполнил работу не полностью или объем выполненной части работы не позволяет сделать правильных выводов. В ответах на вопросы есть грубые ошибки. Нет знания принципиальных теоретических положений темы –1-3 балла

7.2.2. Вопросы к тестовым заданиям

(наименование оценочного средства)

1. Какой тип данных используется для хранения целых чисел в Python?

- a) float
- b) int**
- c) char
- d) number

2. Какой результат вернёт выражение `type("123")`?

- a) int
- b) str**
- c) list
- d) tuple

3. Что произойдёт при выполнении выражения `3 / 2` в Python?

- a) будет выполнено целочисленное деление
- b) будет ошибка типов
- c) результат будет типом float**
- d) результат будет типом int

4. Какой оператор отвечает за возведение в степень?

- a) ^
- b) ****
- c) pow
- d) ^^

5. Какой результат вернёт выражение `len([1, 2, 3, 4])`?

- a) 3
- b) 4**
- c) ошибка
- d) зависит от Python

6. Чем отличается список от кортежа?

- a) списки изменяемые, кортежи нет**
- b) кортежи занимают больше памяти
- c) кортежи поддерживают только числа
- d) списки нельзя перебирать

7. Как создать пустой словарь?

- a) []
- b) ()
- c) {}**
- d) `dict = empty`

8. Какой метод строки разделяет строку на список подстрок?

- a) `break()`
- b) `split()`**
- c) `divide()`
- d) `cut()`

9. Какой тип данных используется для хранения уникальных неупорядоченных значений?

- a) `list`
- b) `set`**
- c) `tuple`
- d) `dict`

10. Какой оператор используется для объединения строк?

- a) `&`
- b) `+`**
- c) `concat`
- d) `merge`

11. Как получить элемент списка по индексу 0?

- a) `list.get(0)`
- b) `list[0]`**
- c) `list.index[0]`
- d) `list(0)`

12. Что делает метод `append()`?

- a) добавляет элемент в начало списка
- b) добавляет элемент в конец списка**
- c) удаляет элемент
- d) копирует список

13. Какой результат вернёт выражение "a" in "apple"?
- a) False
 - b) True**
 - c) None
 - d) ошибка
14. Что произойдёт при попытке изменить элемент кортежа?
- a) значение изменится
 - b) будет создан новый кортеж
 - c) будет ошибка**
 - d) произойдёт преобразование в список
15. Как проверить длину строки?
- a) size(str)
 - b) length(str)
 - c) len(str)**
 - d) count(str)

Критерии оценки за пройденный тест:

40 баллов выставляется обучающемуся, если он ответил правильно на все вопросы рандомной выборки 30 тестовых заданий;

0-39 баллов выставляется обучающемуся в зависимости от количества верных ответов

7.3.1. Вопросы к промежуточной аттестации

№ п/п	Вопросы к экзамену
1.	Перечислите основные типы данных, используемых в Python.
2.	Как задаются числовые типы данных? Какие операции применимы к ним?
3.	Чем отличается список от кортежа?
4.	Какие операции можно осуществлять над строками?
5.	Что такое множества? Какие методы используются при работе с ними?
6.	В чем заключаются особенности создания словаря? Приведите способы создания словарей.
7.	Каким способом можно преобразовать строку в список (кортеж) и обратно?
8.	Каким образом с помощью метода split можно получить список из строки?
9.	Каковы основные особенности структуры программы на Python?
10.	Опишите правила именования переменных. Объясните отличия объявления переменных в Python от других языков программирования.
11.	Что такое инструкции в Python?
12.	Опишите особенности использования функций print() и input().
13.	Поясните назначение метода format и приведите примеры его применения.
14.	Каков синтаксис организации ветвления алгоритма программы?
15.	Как организуются циклы в Python? Перечислите и опишите основные способы.
16.	Как создать пользовательскую функцию и вызвать её в теле программы?
17.	Что такое модули? Перечислите основные модули стандартной библиотеки Python.
18.	Как организовать работу с файлами?
19.	Что такое исключения? Каковы способы их обработки?
20.	Назовите основные типы исключений и укажите причины их

№ п/п	Вопросы к экзамену
	возникновения.
21.	Что такое класс в Python? Каковы его основные характеристики?
22.	Опишите базовые принципы объектно-ориентированного программирования.
23.	Что такое экземпляр класса? Каким образом можно осуществить его создание?
24.	Дайте определение атрибута класса и опишите его основные особенности.
25.	Что такое методы класса? Каковы особенности создания и вызова метода?
26.	Каковы отличия закрытых методов от обычных?
27.	В чем заключается преимущество использования конструктора <code>__init__()</code> при создании класса?
28.	Как осуществляется перегрузка специальных методов класса в Python?
29.	Как реализуется принцип наследования в Python? Приведите примеры.
30.	В чем смысл использования абстрактного метода в Python?
31.	Что такое событийно-ориентированное программирование?
32.	Дайте определение понятий «событие», «обработчик событий» и «цикл обработки событий».
33.	Перечислите основные инструменты для создания графического интерфейса.
34.	Опишите алгоритм построения интерфейса на базе главного окна и способы размещения виджетов.
35.	Каковы синтаксис создания главного окна и его основные атрибуты?
36.	Опишите особенности построения текстовых виджетов.
37.	Каковы особенности создания управляющих кнопок?
38.	Как создать меню?
39.	Как реализовать связывание событий, инициированных пользователем, с обработчиком этих событий? Опишите синтаксис соответствующего метода.
40.	Опишите особенности построения виджета <code>canvas</code> и работу с основными графическими примитивами.
41.	Что такое идентификаторы в Python? Каковы правила их формирования?
42.	Дайте определение понятию «выражение» в Python. Какие виды выражений существуют?
43.	Что такое операторы присваивания? Перечислите расширенные операторы присваивания.
44.	Объясните отличие <code>mutable</code> и <code>immutable</code> типов данных. Приведите примеры.
45.	Как работает механизм интернирования строк в Python?
46.	Как реализуется проверка вхождения элемента в коллекцию? Приведите примеры для списка, множества и словаря.
47.	Что такое диапазоны (<code>range</code>)? Перечислите способы их создания и особенности.
48.	Опишите механизм срезов (<code>slice</code>). Для каких типов он применим, как работает?
49.	Объясните особенности хранения списков в памяти (массив указателей).
50.	Что такое генераторы списков? Приведите примеры и объясните их преимущества.
51.	Что такое <code>lambda</code> -функции? Каковы ограничения и области применения?
52.	Объясните разницу между функциями <code>map()</code> , <code>filter()</code> , <code>reduce()</code> . Приведите примеры.
53.	Что такое рекурсия? Какие условия обязательны для рекурсивной функции?

№ п/п	Вопросы к экзамену
54.	Дайте определение области видимости переменных. Что такое LEGB-модель?
55.	Что такое замыкание? Приведите пример.
56.	Чем отличаются функции от процедур в контексте Python?
57.	Объясните механизм передачи аргументов по ссылке/значению в Python.
58.	Что такое *args и kwargs? Каковы правила их использования?
59.	Какие существуют режимы открытия файлов? Объясните разницу между 'w', 'a', 'x'.
60.	Чем отличается текстовый режим работы с файлами от бинарного?
61.	Что такое контекстный менеджер и оператор with? Приведите пример.
62.	Объясните разницу между исключениями уровня интерпретатора и пользовательскими исключениями.
63.	Что такое конструкция try-except-else-finally? Когда целесообразно использовать блок else?
64.	Как создать собственный класс исключения?
65.	Что такое атрибуты класса и атрибуты экземпляра? Каковы различия в хранении данных?
66.	Объясните назначение параметра self. Почему он обязателен?
67.	Что такое статические методы и методы класса? Приведите примеры.
68.	Опишите особенности механизма MRO (Method Resolution Order).
69.	Что такое множественное наследование? Приведите пример и опишите возможные сложности.
70.	Поясните механизм инкапсуляции в Python и условность закрытых атрибутов.
71.	Что такое переопределение методов (override)? Когда оно используется?
72.	Что такое полиморфизм? Приведите пример в контексте Python.
73.	Зачем используется метод str() и чем он отличается от repr()?
74.	Что такое свойства (property)? Как их создавать и для чего применять?
75.	Объясните разницу между функцией и методом в Python.
76.	Что такое модульная структура программы? Каковы преимущества модульности?
77.	Что представляет собой пакет (package) в Python? Какова роль файла init.py?
78.	Как работает импорт модулей? Объясните отличия import, from ... import, import as.
79.	Какие существуют принципы проектирования классов и модулей (DRY, KISS, cohesion)?
80.	Что такое тестирование программ? Перечислите основные виды тестов и применимость unittest в Python.

7.3.2. Критерии и нормы оценки

Семестр	Форма проведения промежуточной аттестации	Критерии и нормы оценки	
3	Экзамен (по накопительному рейтингу)	отлично	от 85 до 100 баллов
		хорошо	от 75 до 84 баллов
		удовлетворительно	от 55 до 74 баллов

Семестр	Форма проведения промежуточной аттестации	Критерии и нормы оценки	
		неудовлетворительно	менее 55 баллов

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Обязательная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно- методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1.	Гуриков С.Р.	Основы алгоритмизации и программирования на Python	Учебное пособие	2022	ЭБС «Znanium»
2	Жуков Р.А.	Язык программирования Python: практикум	Учебное пособие	2022	ЭБС «Znanium»

8.2. Дополнительная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно- методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1.	Дроботун Н.В.	Алгоритмизация и программирование. Язык Python	Учебное пособие	2020	ЭБС «IPRBooks»
2.	Широков А.И.	Информатика: разработка программ на языке программирования Питон: базовые языковые конструкции	Учебник	2020	ЭБС «IPRBooks»

8.3. Перечень профессиональных баз данных и информационных справочных систем

№ пп	Наименование	Ссылка
1	Springer Nature (Полнотекстовая коллекция журналов)	https://www.springernature.com/gp/products
2	Springer eBooks (Полнотекстовая коллекция электронных книг издательства Springer Nature)	https://link.springer.com/
3	«Кодекс»	https://kodeks.ru/
4	ELIBRARY.RU (электронная библиотека научных публикаций)	http://elibrary.ru
5	"Гарант"	https://www.garant.ru/
6	"КонсультантПлюс"	https://www.consultant.ru/
7	Техэксперт	https://cntd.ru/

8.4. Перечень программного обеспечения

№ п/п	Наименование ПО	Реквизиты договора (дата, номер, срок действия)
1.	WinPro 10 RUS Upgrd OLP NL Acdmc	Договор № 757 от 04.07.2018, срок действия – бессрочно; Контракт № 1653 от 14.12.2018, срок действия – бессрочно
2	Office Stdandard 2013 Russian OLP NL AcademicEdition	Контракт № 690 от 19.05.2015, срок действия – бессрочно
3	Python 3.11	Лицензия Python Software Foundation License (PSFL)
4	Jupyter Notebook	Лицензия: Modified BSD License (Open Source).
5	Google Colab	Лицензия: Бесплатный облачный сервис на базе открытого ПО.

8.5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)	Перечень основного оборудования
1	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для проведения лабораторных работ. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации. (УЛК-402)	Системные блоки (HP ProDesk), мониторы (Samsung), коммутатор (D-Link), столы ученические, столы компьютерные, стулья, доска аудиторная.
2	Помещение для самостоятельной работы обучающихся. (УЛК-105)	Стол, стулья, стеллажи (в т.ч. выставочные) с книгами, компьютеры, мобильные рабочие места.
3	Помещение для самостоятельной работы обучающихся. (УЛК-406)	Стол компьютерный, стулья, микрокомпьютеры raspberry pi 32 bit.