

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»
(БГТУ им. В.Г. Шухова)



РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)

Программирование

Направление подготовки:
38.03.05 – Бизнес-информатика

Направленность программы (профиль):
Технологическое предпринимательство

Квалификация
бакалавр

Форма обучения
очная

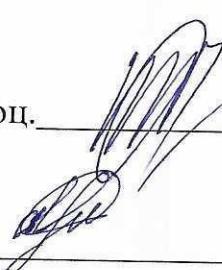
Институт информационных технологий и управляемых систем
Кафедра прикладной информатики

Белгород 2025

Рабочая программа составлена на основании требований:

▪ Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 38.03.05 – Бизнес-информатика, утвержденного приказом Министерства науки и высшего образования Российской Федерации от 29.07.2020 № 838;

▪ Учебного плана, утвержденного ученым советом БГТУ им. В. Г. Шухова в 2021 году.

Составитель (составители): канд. экон. наук, доц.  (С.П. Гавриловская)

Составитель (составители): ст. препод.  (Р.А. Мясоедов)

Рабочая программа обсуждена на заседании кафедры экономики и организации производства

« 28 » апреля 2025 г., протокол № 5

Заведующий кафедрой экономики и организации производства

канд. экон. наук, доц.  (Д.В. Кадацкая)

Рабочая программа согласована с выпускающей кафедрой экономики и организации производства

Заведующий кафедрой: канд. экон. наук, доц.  (Д.В. Кадацкая)

« 28 » апреля 2025 г.

Рабочая программа одобрена методической комиссией института

« 29 » апреля 2025 г., протокол № 8

Председатель доц.  (Ю.Д. Рязанов)

1. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Наименование показателя оценивания результата обучения по дисциплине
ОПК-3. Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3.2 Разрабатывает алгоритмы, выполняет отладку и тестирование программ с целью применения их в сфере информационно-коммуникационных технологий	<p>Знания:</p> <ul style="list-style-type: none"> – основных принципов, методов и понятий структурного программирования; – теоретических основ конструирования алгоритмов и их применения при решении задач в сфере информационно-коммуникационных технологий; – основ языка программирования Python, включая типы данных, управляющие структуры, операторы, стандартные библиотеки функций и базовые конструкции; <p>Умения:</p> <ul style="list-style-type: none"> – выполнение обоснования использования алгоритмических конструкций при решении задачи; – выполнение реализации алгоритма решения задачи в виде программного кода в среде PyCharm. <p>Навыки:</p> <ul style="list-style-type: none"> – применение методов для составления алгоритма решения задачи; – использование языка программирования для практической реализации алгоритма решения задачи; – применение инструментария для выполнения отладки и тестирования программного кода.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1. Компетенция ОПК-3. Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации

Данная компетенция формируется следующими дисциплинами:

Стадия	Наименования дисциплины
1	Информационные технологии в экономике и управлении
2	Программирование
3	Базы данных

3. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 7 зач. единиц, 252 часа.
Форма промежуточной аттестации зачет, дифференцированный зачет

Вид учебной работы	Всего часов	Семестр № 2	Семестр № 3
Общая трудоемкость дисциплины, час	252	108	144
Контактная работа (аудиторные занятия), в т.ч.:	124	53	71
лекции	51	17	34
лабораторные	68	34	34
практические			
групповые консультации в период теоретического обучения и промежуточной аттестации	5	2	3
Самостоятельная работа студентов, включая индивидуальные и групповые консультации, в том числе:	128	55	73
Курсовой проект			
Курсовая работа			
Расчетно-графическое задание	18		18
Индивидуальное домашнее задание	9	9	
Самостоятельная работа на подготовку к аудиторным занятиям (лекции, практические занятия, лабораторные занятия)	101	46	55
Дифференцированный зачет, зачет		Зачет	Диф. зачет

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Наименование тем, их содержание и объем Курс 1 Семестр 2

№ п/п	Наименование раздела (краткое содержание)	Объем на тематический раздел по видам учебной нагрузки, час			
		Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа на подготовку к аудиторным занятиям
1	2	3	4	5	6
1. Основные принципы алгоритмизации и программирования					
1.	Основные понятия алгоритмизации. Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов. Общие принципы построения алгоритмов. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические	3		4	6
2.	Языки и системы программирования. Технология программирования. Принципы структурного и модульного программирования. Эволюция и	3		4	6

	классификация языков программирования. Понятие системы программирования			
2. Программирование на языке Python				
1	Основные элементы языка. Лексика языка. Переменные и константы. Типы данных. Выражения и операции.	2	6	8
2	Операторы языка. Синтаксис операторов: присваивания, ввода-вывода. Синтаксис операторов: безусловного и условного переходов. Синтаксис операторов циклов. Составной оператор. Вложенные условные операторы. Циклические конструкции. Циклы с предусловием и постусловием, с параметром.	5	10	14
3	Организация ввода-вывода данных. Работа с файлами. Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа. Запись в файл и чтение из файла последовательного доступа. Стандартные процедуры и функции для файлов разного типа	4	10	12
	ВСЕГО	17	34	46

Курс 2 Семестр 3

№ п/п	Наименование раздела (краткое содержание)	Объем на тематический раздел по видам учебной нагрузки, час			
		Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа на подготовку к аудиторным занятиям
1	2	3	4	5	6
2. Программирование на языке Python					
4.	Массивы. Массивы, как структурированный тип данных. Объявление массива. Ввод и вывод одномерных массивов. Ввод и вывод двухмерных массивов. Обработка массивов. Стандартные функции для массива целых и вещественных чисел.	12		14	22
5.	Строки и множества. Структурированные типы данных: строки и множества. Объявление строковых типов данных. Поиск, удаление, замена и добавление символов в строке. Операции со строками. Стандартные функции и процедуры для работы со строками. Объявление множества. Операции над множествами	10		12	18
6	Процедуры и функции. Понятие подпрограммы. Процедуры и функции, их сущность, назначение, различие. Организация процедур, стандартные процедуры. Процедуры, определенные пользователем: синтаксис, передача аргументов. Формальные и фактические параметры. Процедуры с параметрами, описание процедур. Функции: способы организации и описание. Вызов функций, рекурсия. Программирование рекурсивных алгоритмов.	12		8	15

	Стандартные функции.			
	ВСЕГО	34	34	55

4.2. Содержание практических (семинарских) занятий

Практические (семинарские) работы не предусмотрены планом учебного процесса.

4.3 Содержание лабораторных занятий

№ п/п	Наименование раздела дисциплины	Тема лабораторного занятия	К-во часов	К-во часов СРС
семестр №2				
1	Основные принципы алгоритмизации и программирования	Лабораторная работа №1 Составление блок-схем линейных, разветвляющихся, циклических алгоритмов	4	4
		Лабораторная работа №2 Среда программирования PyCharm	4	4
2	Программирование на языке Python	Лабораторная работа №3 Программирование алгоритмов линейной структуры	6	6
		Лабораторная работа №4 Программирование алгоритмов разветвляющейся структуры	4	4
		Лабораторная работа №5 Программирование итерационных алгоритмов циклической структуры	6	6
		Лабораторная работа №6 Работа с файлами	10	10
		ИТОГО:		34
		ВСЕГО:		68

№ п/п	Наименование раздела дисциплины	Тема лабораторного занятия	К-во ча- сов	К-во часов СРС
семестр №3				
2	Программирование на языке Python	Лабораторная работа №7 Программирование с использованием одномерных массивов	6	6
		Лабораторная работа №8 Программирование с использованием двумерных массивов	8	8
		Лабораторная работа №9 Работа со строковыми переменными	6	6
		Лабораторная работа №10 Программирование с использованием множеств	6	6
		Лабораторная работа №11 Программирование с использованием процедур и функций	8	8
		ИТОГО:		34
ВСЕГО:				
				68

4.4. Содержание курсового проекта/работы

Курсовой проект или работа не предусмотрены учебным планом.

4.5. Содержание расчетно-графического задания, индивидуальных домашних заданий

В процессе выполнения индивидуального домашнего задания осуществляется контактная работа обучающегося с преподавателем. Консультации проводятся в аудитории и/или посредством электронной информационно-образовательной среды университета.

На выполнение ИДЗ предусмотрено 9 часов самостоятельной работы студента.

Цель задания: является освоение методологии структурного программирования. Разработка программного комплекса направлена на приобретение студентом практических навыков анализа задачи, ее декомпозиции, работы с различными структурами данных, типовыми алгоритмами, подпрограммами.

Студент должен разобраться с поставленной задачей, отладить ее на персональном компьютере и продемонстрировать преподавателю работу отлаженных программ в соответствии с вариантом задания.

Каждый студент должен разработать простейшую программу, используя технологию файловых систем. При этом должны быть созданы файлы данных, реализованы задачи дополнения, корректировки и обработки данных, в соответствии с заданием выполнено графическое представление результатов обработки информации.

Структура работы. При выполнении индивидуального домашнего задания (ИДЗ) студент должен обеспечить выполнение следующих требований:

1. В работе должен быть предусмотрен пользовательский интерфейс, организовано меню решаемых задач.

2. Ввод данных в файл и запросов пользователя должен сопровождаться проверкой ввода.

3. Согласно индивидуальному заданию должна быть построена блок-схема алгоритма одной из программ.

4. Должна быть разработана заставка системы, содержащая название программы, информацию об ее назначении и фамилию автора разработки.

5. Работа заканчивается отчетом, содержащим постановку задачи, структуру всей подсистемы, описание каждой части подсистемы, инструкцию по работе с подсистемой.

На защите ИДЗ студент должен продемонстрировать работоспособность программы на тестовом примере.

Примерные варианты заданий

Задание 1

Написать программы для решения предложенных ниже задач. Пользователь вводит допустимое значение x , программа выдает результат.

$$y = (1 + \sin x)^2 \quad y = \sqrt{1 + x^2} \quad y = |5 - 6x| \quad y = \log_2 \frac{x^2 + 1}{3}$$

$$y = \sqrt[4]{1 + x + x^2} \quad y = \sin(\cos x) \quad y = \operatorname{ctg}(x - 4) \quad y = 2^{x+1}$$

Задание 2

1. Вычислить длину окружности, площадь круга и объем шара одного и того же радиуса

2. По координатам трех вершин некоторого треугольника найти его периметр и площадь

3. Вычислить периметр и площадь прямоугольного треугольника по длинам двух катетов

4. Вычислить площадь полной поверхности и объем прямого кругового цилиндра, заданного высотой и диаметром основания.

5. Вычислить площадь полной поверхности и объем прямого кругового цилиндра, заданного высотой и диаметром основания.

6. Вычислить периметр и площадь правильного 15-угольника, вписанного в окружность заданного радиуса

7. Вычислить площадь полной поверхности и объем правильного тетраэдра по длине его ребра.

8. Вычислить площадь параллелограмма, у которого даны длины сторон и величина угла между ними (в градусах).

Задание 3

1. Найти максимум из трех заданных целых чисел a, b, c

2. Даны произвольные целые числа a, b, c . Определить, можно ли построить треугольник с длинами сторон, равными этим числам.

3. Найти действительные корни квадратного уравнения $ax^2 + bx + c = 0$. Если корней нет, то вывести сообщение об этом.

4. Решить систему уравнений

$$y = \begin{cases} ax + by = c \\ dx + ey = f \end{cases}$$

5. Данна точка $A(x, y)$. Определить, какой из координатных четвертей она принадлежит.

6. Определить, какая из трех точек $A(x_1, y_1), B(x_2, y_2), C(x_3, y_3)$ ближе к началу координат.

7. Если треугольник со сторонами a, b, c является прямоугольным, то напечатать длину медианы, выходящей из прямого угла, в противном случае напечатать его периметр.

8. Если треугольник со сторонами a, b, c является равнобедренным, найти его площадь, в противном случае найти его периметр.

В процессе выполнения расчетно-графического задания осуществляется контактная работа обучающегося с преподавателем. Консультации проводятся в аудитории и/или посредством электронной информационно-образовательной

среды университета.

На выполнение РГЗ предусмотрено 18 часов самостоятельной работы студента.

Цель задания: приобретение студентом практических навыков проектирования, разработки, отладки и документирования программ среднего уровня сложности или небольших программных систем.

РГЗ является одним из этапов изучения дисциплины «Программирование», который позволяет судить об усвоении студентом теоретического курса, использовании знаний для решения конкретных задач по программированию. Задание для выполнения РГЗ выдается преподавателем. При желании студенты могут предложить свою тему для разработки программы. В этом случае задание должно быть представлено в письменной форме с четкой формулировкой (содержательной и математической) существа задачи, описания входных и выходных данных, требованиями по оформлению готовой программы и документации, списком рекомендуемой литературы и рассмотрено на заседании кафедры.

Структура работы.

При выполнении расчетно-графического задания (РГЗ) студент должен обеспечить выполнение следующих требований:

- подготовить содержательную и формализованную постановку задачи;
- составить обоснование выбранного метода решения задачи и работы системы;
- начертить блок-схемы алгоритмов для каждого уровня детализации задачи с необходимыми пояснениями;
- написать техническое задание;
- составить тексты программ;
- оформить руководство пользователя;
- подготовить контрольный пример, демонстрирующий правильность выполнения программы в различных режимах ее функционирования и с различными вариантами типов исходных данных;
- составить список используемой литературы.

Все перечисленные материалы должны быть в рабочем состоянии и по содержанию соответствовать друг другу в отлаженной программе.

Примерные варианты заданий для выполнения РГЗ

1. Разработка системы расчета прибыли цветочного салона от реализации продукции за 1 месяц.
2. Разработка системы расчета учета посещений и оплаты в детском саду.
3. Разработка системы расчета прибыли салона красоты от услуг, оказываемых клиентам.
4. Разработка системы расчета прибыли агентства по продаже недвижимости.
5. Разработка системы расчета прибыли мебельной фабрики от реализации продукции.
6. Разработка системы расчета прибыли отдела по продаже видеотехники универмага.
7. Разработка системы учета денежных вкладов клиентов банка.

8. Разработка системы расчета прибыли магазина от продажи товаров.
9. Разработка системы расчета прибыли кондитерской фабрики от реализации продукции.
10. Разработка системы расчета прибыли телерадиокомпании от рекламной деятельности.
11. Разработка системы расчета прибыли диагностического центра от услуг, оказываемых населению.
12. Разработка системы расчета прибыли станции технического обслуживания автомобилей.
13. Разработка системы расчета калорийности блюд в столовой.
14. Разработка системы расчета ежедневной прибыли КПП таможни.
15. Разработка системы учета продаж билетов в пароходной компании.

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

5.1. Реализация компетенций

1. Компетенция ОПК-3. Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации.

Наименование индикатора достижения компетенции	Используемые средства оценивания
ОПК-3.2 Разрабатывает алгоритмы, выполняет отладку и тестирование программ с целью применения их в сфере информационно-коммуникационных технологий	Зачет, дифференцированный зачет, защита ИДЗ, защита РГЗ, защита лабораторных работ, собеседование

5.2. Типовые контрольные задания для промежуточной аттестации

5.2.1. Перечень контрольных вопросов (типовых заданий) для зачета

№ п/п	Наименование раздела дисциплины	Содержание вопросов (типовых заданий)
1	Основные принципы алгоритмизации и программирования	<p>1. Что такое система счисления?</p> <p>2. Как представляются числа в форме с фиксированной и плавающей запятой? Что такое бит, байт, килобайт, мегабайт и гигабайт?</p> <p>3. Что такое: алгоритм, машинная программа, машинная команда? Каковы структура и назначение кодов ASCII?</p> <p>4. Что такое система классификации информации? Что такое система кодирования информации? Классификация методов.</p> <p>5. Перечислите способы записи алгоритма.</p> <p>6. Что такое язык программирования?</p> <p>7. В чем различие компиляторов и интерпретаторов?</p> <p>8. Расскажите о поколениях языков программирования.</p> <p>9. Что такое модульное программирование?</p> <p>10. Что такое структурное программирование?</p> <p>11. Что такое жизненный цикл программы?</p> <p>12. Что подразумевается под средой программирования?</p>

		<p>13. Охарактеризуйте назначение окон, команд, меню и панелей среды разработки программ.</p> <p>14. Как выполнить компиляцию и запуск приложения в среде программирования?</p> <p>15. Какие символы входят в состав алфавита языка программирования?</p> <p>16. Для чего используются идентификаторы? По каким правилам образуются идентификаторы?</p>
2	Программирование на языке Python	<p>17. Что такое переменная? В каком месте программы описываются (объявляются) переменные? Что такое константа?</p> <p>18. Какие существуют типы данных?</p> <p>19. Что представляет собой выражение? Что такое оператор (инструкция)? Для чего и как используются комментарии?</p> <p>20. Как выглядит оператор (инструкция) присваивания?</p> <p>21. Как выполняются ввод и вывод данных?</p> <p>22. Какие выражения используются в качестве условий в условных операторах (инструкциях)?</p> <p>23. Как выполняется условный оператор (инструкция)?</p> <p>24. Как выполняется сокращенная форма условного оператора (инструкции)?</p> <p>25. Какие разновидности циклов существуют?</p> <p>26. В каком направлении происходит переход при организации цикла с использованием условных операторов (инструкций) и операторов (инструкций) перехода?</p> <p>27. Как выполняется оператор цикла с предусловием?</p> <p>28. Как выполняется оператор цикла с постусловием?</p> <p>29. Как выполняется оператор цикла с параметром?</p> <p>30. Какие компоненты предназначены для работы с дисками и каталогами?</p> <p>31. Какие компоненты предназначены для открытия и сохранения файлов?</p> <p>32. Какие подпрограммы используются для работы с файлами?</p> <p>33. В чем состоит сущность объектной модели работы с файлами?</p>

Перечень контрольных вопросов (типовых заданий) для дифференцированного зачета

№ п/п	Наименование раздела дисциплины	Содержание вопросов (типовых заданий)
1	Программирование на языке Python	<p>1. Что такое массив?</p> <p>2. Как располагаются данные в многомерных массивах?</p> <p>3. Как описываются (объявляются) одномерные массивы?</p> <p>4. Как можно выполнить ввод данных в массив?</p> <p>5. Как можно выполнить вывод данных из массива?</p> <p>6. Как указываются количества позиций для вывода целых и вещественных чисел при использовании форматов вывода?</p> <p>7. Какая разновидность оператора цикла чаще всего используется при работе с массивами?</p> <p>8. Как выглядит тело оператора цикла для суммирования элементов одномерного массива?</p> <p>9. Как выглядит оператор цикла для суммирования элементов двумерного массива?</p> <p>10. Как обычно называется цикл, оператор которого является телом другого цикла?</p>

	<p>11. Как обычно называется цикл, телом которого также является оператор цикла?</p> <p>12. Как описываются (объявляются) динамические массивы?</p> <p>13. Как в программе происходит использование динамического массива?</p> <p>14. Как описываются (объявляются) многомерные массивы?</p> <p>15. Как указываются количества позиций для вывода целых и вещественных чисел при использовании форматов вывода?</p> <p>16. Как указываются количества позиций для вывода целых и вещественных чисел при использовании форматов вывода?</p> <p>17. Как выглядит оператор цикла для суммирования элементов двумерного массива?</p> <p>18. Как обычно называется цикл, оператор которого является телом другого цикла?</p> <p>19. Что такое подпрограмма?</p> <p>20. Что представляют собой процедуры и как они используются?</p> <p>21. Где в программе располагается описание (объявление) процедуры?</p> <p>22. Как в программе записывается обращение к процедуре?</p> <p>23. Что представляют собой функции и как они используются?</p> <p>24. Где в программе располагается описание (объявление) функции?</p> <p>25. Как в программе записывается обращение к функции?</p> <p>26. Как называются параметры процедуры или функции, используемые при ее описании (объявлении)?</p> <p>27. Как называются параметры процедуры или функции, используемые при обращении к ней?</p> <p>28. Какие операции можно выполнять с символами и строками?</p> <p>29. Какие типы данных могут определяться пользователем и как эти данные могут использоваться?</p> <p>30. Что такое запись?</p> <p>31. Какие структуры данных относятся к динамическим?</p> <p>32. Каковы основные виды списков и способы их реализации?</p>
--	--

5.2.2. Перечень контрольных материалов для защиты курсового проекта/курсовой работы

Курсовой проект или работа не предусмотрены учебным планом.

5.2.3. Типовые контрольные задания (материалы) для текущего контроля в семестре

Текущий контроль осуществляется в течение всего периода изучения дисциплины в форме выполнения и защиты лабораторных работ, выполнения контрольных работ, тестового контроля, выполнения индивидуального домашнего задания, выполнения и защиты расчетно-графического задания.

Лабораторные работы. В лабораторном практикуме по дисциплине представлен перечень лабораторных работ, обозначены цели и задачи, даны необходимые теоретические и методические указания к работе, рассмотрен практический пример, приведены варианты выполнения и перечень контрольных вопросов.

Защита лабораторных работ возможна после проверки правильности выполнения задания, оформления отчета. Защита проводится в форме

собеседования преподавателя со студентом по теме лабораторной работы. Примерный перечень контрольных вопросов для защиты лабораторных работ представлен в таблице.

№	Тема лабораторной работы	Контрольные вопросы
1.	Лабораторная работа №1. Составление блок-схем линейных, разветвляющихся, циклических алгоритмов.	<ol style="list-style-type: none"> Свойства алгоритма. Типы вычислительных процессов. Перечислите основные элементы блок-схем и их назначение. Какие блоки используются при реализации линейного, разветвляющегося, циклического алгоритмов? Можно ли составить разные варианты блок-схем для одной и той же задачи? Какие виды циклического алгоритма бывают? Какие пункты должны присутствовать в любом цикле?
2.	Лабораторная работа №2. Среда программирования Python.	<ol style="list-style-type: none"> С чего начинается программа? Как описываются переменные? Как происходит присваивание переменной какого-либо значения? Как называется этот оператор? Чем заканчивается программа? Как сохранить программу на диске? Как сохранить программу под другим именем? Как открыть новый файл? Как открыть уже существующий файл? Какие имена можно использовать в языке Python? Для чего необходим в программе раздел описаний? Какие стандартные типы вам известны? В чем различия между данными действительного и целого типов?
3.	Лабораторная работа №3. Программирование алгоритмов линейной структуры.	<ol style="list-style-type: none"> Как записываются операторы начала и конца программы? Из каких разделов состоит программа на языке Python? В какой последовательности должны быть записаны разделы программы на языке Python? Как записываются операторы вывода на экран в Python? Как записываются операторы ввода в Python? Как можно его использовать, чтобы осуществить задержку экрана? Какие обозначения используются в форматах вывода?
4.	Лабораторная работа №4. Программирование алгоритмов разветвляющейся структуры.	<ol style="list-style-type: none"> Какие операторы используются для программирования разветвлений? Как выполняются операторы перехода? Для чего служит ключ выбора и какого он может быть типа? Особенности использования вложенных условных операторов. Каковы отличия оператора выбора Case от оператора условия if? Какую из функций: Sin(x), Abs(x), Trunc(x) можно заменить условным оператором if x < 0 then x := -x?
5.	Лабораторная работа №5. Программирование итерационных алгоритмов циклической структуры.	<ol style="list-style-type: none"> Как записывается и как работает оператор FOR? Для организации каких циклов применим оператор FOR? В чем отличие оператора WHILE от оператора REPEAT?

		<p>4. Как программируются циклические алгоритмы с явно заданным числом повторений цикла?</p> <p>5. Как программируются циклические алгоритмы с заранее неизвестным числом повторений цикла?</p> <p>6. Напишите оператор цикла, который не выполняется ни разу.</p> <p>7. Напишите оператор цикла, который выполняется неограниченное число раз.</p>
6	Лабораторная работа №6. Работа с файлами.	<p>1. Что такое файл? Какие существуют виды файлов?</p> <p>2. Какими стандартными процедурами и функциями располагает Python для работы с файлами?</p> <p>3. Каково должно быть содержание программы по созданию файла и его корректировки (замена элементов, добавление элементов, удаление элементов)?</p> <p>4. Каковы особенности работы с текстовыми файлами?</p> <p>5. Каковы особенности работы с типизированными файлами?</p> <p>6. Как подсчитать число строк в текстовом файле?</p>
7	Лабораторная работа №7. Программирование с использованием одномерных массивов.	<p>1. Что такое массив? Почему массив является структурированным типом данных?</p> <p>2. Что такое размерность массива? Существуют ли ограничения на размерность массива?</p> <p>3. Какого типа могут быть элементы массива? Какого типа могут быть индексы элементов массива?</p> <p>4. Какие простые типы данных относятся к порядковым?</p> <p>5. Какими способами может быть заполнен массив? Приведите примеры.</p> <p>6. Какими способами можно вывести на экран массив? Приведите примеры.</p> <p>7. Какие действия выполняют обычно над элементами массива?</p>
8	Лабораторная работа №8. Программирование с использованием двумерных массивов.	<p>1. Что такое двумерный массив?</p> <p>2. Как определяется размерность двумерного массива?</p> <p>3. Какого типа могут быть элементы массива?</p> <p>4. Какого типа могут быть индексы элементов массива?</p> <p>5. Какими способами может быть заполнена матрица? Приведите примеры.</p> <p>6. Какими способами можно вывести на экран матрицу? Приведите примеры.</p> <p>7. Какие действия выполняют обычно над элементами матрицы?</p>
9	Лабораторная работа №9. Работа со строковыми переменными.	<p>1. Как описываются строковые переменные?</p> <p>2. Какая максимальная длина строки допустима в Python?</p> <p>3. Какие операции допустимы над строковыми данными?</p> <p>4. В чем отличие строковой переменной от массива символов?</p> <p>5. Какие стандартные процедуры и функции для работы со строками вы знаете?</p>
10	Лабораторная работа №10. Программирование с использованием множеств.	<p>1. Что такое множество, как оно описывается в Python?</p> <p>2. Как определить новый тип данных с использованием перечисления? Как описываются типизированные константы типа множество?</p> <p>3. Как осуществляется ввод-вывод значений переменных типа множество?</p>

		<ol style="list-style-type: none"> 4. Какие типы данных используются в качестве базовых при объявлении типа множество? 5. Какие операции определены над множествами? 6. Какие операции допустимы над переменными, заданными перечислением? 7. Чем похожи и чем отличаются множества и массивы?
11	Лабораторная работа №11. Программирование с использованием процедур и функций.	<ol style="list-style-type: none"> 1. Для чего нужны в программе процедуры и функции? 2. В чем отличие между процедурой и функцией? 3. Чем отличаются формальные и фактические параметры? 4. Чем отличаются параметры-значения и параметры-переменные? 5. Как объявляются глобальные и локальные переменные? Каково правило их видимости? 6. Почему при обращении к процедуре, аргумент, передаваемый параметру-переменной, может быть только переменной, а не константой или выражением?

Индивидуальное домашнее задание.

Индивидуальное домашнее задание является формой самостоятельной работы обучающегося. Решение ИДЗ выполняется студентами самостоятельно по заданиям, выдаваемым преподавателем. В процессе выполнения индивидуального домашнего задания осуществляется контактная работа обучающегося с преподавателем. Защита ИДЗ происходит в форме собеседования преподавателя и студента по представленному в ней материалу. Обучающемуся могут быть заданы вопросы по материалам изучаемой дисциплины.

Оформление индивидуального домашнего задания. Индивидуальное домашнее задание предоставляется преподавателю для проверки в двух видах: отчет, на бумажных листах в формате А4, и в виде файлов, содержащих разработанные в соответствии с вариантом программы. Отчет индивидуального домашнего задания должен иметь следующую структуру:

Титульный лист;

Оглавление;

1. Задание на разработку программного комплекса;

2. Функциональная структура приложения;

3. Спецификация программного комплекса;

4. Структурная диаграмма программного комплекса;

5. Блок-схема алгоритма одного из модулей (в соответствии с заданием);

6. Данные для тестирования программных модулей и программного комплекса в целом;

7. Инструкция для пользователя по работе с программным комплексом;

Приложения: исходные тексты программных модулей.

Срок сдачи ИДЗ определяется преподавателем.

Расчетно-графическое задание является формой самостоятельной работы обучающегося. Решение расчетно-графического задания выполняется студентами самостоятельно по заданиям, выдаваемым преподавателем. Срок сдачи РГЗ определяется преподавателем. Защита РГЗ происходит в форме собеседования преподавателя и студента по представленному в ней материалу. Обучающемуся могут быть заданы вопросы по материалам изучаемой дисциплины.

Оформление расчетно-графического задания. РГЗ предоставляется

преподавателю для проверки в виде отчета и в виде файлов, содержащих решение практических заданий. Отчет расчетно-графического задания должен иметь следующую структуру: титульный лист; содержание; постановка задачи; обоснование метода решения задачи; блок-схемы алгоритмов; техническое задание; тексты программ; руководство пользователя; контрольный пример; список использованной литературы.

5.3. Описание критериев оценивания компетенций и шкалы оценивания

При промежуточной аттестации в форме зачета в семестре №2 используется следующая шкала оценивания: зачтено - не зачтено.

Критериями оценивания достижений показателей являются:

Наименование показателя оценивания результата обучения по дисциплине	Критерий оценивания
ОПК-3. Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	
ОПК-3.2 Разрабатывает алгоритмы, выполняет отладку и тестирование программ с целью применения их в сфере информационно-коммуникационных технологий	
Знания	<p>Знание терминов, стандартов, методов алгоритмизации и структурного программирования</p> <p>Объем освоенного материала</p> <p>Полнота ответов на вопросы</p> <p>Четкость изложения и интерпретации знаний</p>
Умения	<p>Полнота, качество выполненного задания</p> <p>Умение обосновать использование алгоритмов линейной, разветвляющейся и циклической структуры для решения задачи</p> <p>Умение использовать принципы работы с файловым типом данных при решении задачи</p> <p>Умение соотнести полученный результат с поставленной целью</p>
Навыки	<p>Практическая реализация алгоритмов линейной, разветвляющейся и циклической структуры с помощью среды PyCharm</p> <p>Применение файлов для организации ввода-вывода данных в среде PyCharm</p> <p>Анализ результатов выполненных заданий</p>

Оценка преподавателем выставляется интегрально с учетом всех показателей и критериев оценивания.

Оценка сформированности компетенций по показателю **Знания**.

Критерий	Уровень освоения и оценка			
	не зачтено	зачтено		
Знание терминов, стандартов, методов алгоритмизации и структурного программирования	Не знает терминов, стандартов, методов алгоритмизации и структурного программирования	Знает термины, стандарты, методы алгоритмизации и структурного программирования, но допускает неточности формулировок	Знает термины, стандарты, методы алгоритмизации и структурного программирования	Знает термины, стандарты, методы алгоритмизации и структурного программирования, может корректно сформулировать их самостоятельно

Объем освоенного материала	Не знает значительной части материала дисциплины	Знает только основной материал дисциплины, не усвоил его деталей	Знает материал дисциплины в достаточном объеме	Обладает твердым и полным знанием материала дисциплины, владеет дополнительными знаниями
Полнота ответов на вопросы	Не дает ответы на большинство вопросов	Дает неполные ответы на все вопросы	Дает ответы на вопросы, но не все - полные	Дает полные, развернутые ответы на поставленные вопросы
Четкость изложения и интерпретации знаний	Излагает знания без логической и последовательности. Неверно излагает и интерпретирует знания	Излагает знания с нарушениями в логической последовательности. Допускает неточности в изложении и интерпретации знаний	Излагает знания без нарушений в логической и последовательности. Грамотно и по существу излагает знания	Излагает знания в логической последовательности самостоятельно их интерпретируя и анализируя. Грамотно и точно излагает знания, делает самостоятельные выводы

Оценка сформированности компетенций по показателю **Умения**.

Критерий	Уровень освоения и оценка			
	не зачтено	зачтено		
Полнота, качество выполненного задания	Задание не выполнено или выполнено некачественно	Задание выполнено с незначительными ошибками в полном объеме и качественно	Задание выполнено в полном объеме и качественно	Задание выполнено в полном объеме. Обучающимся сформулированы самостоятельные выводы, выполнен анализ полученных
Умение обосновать использование алгоритмов линейной, разветвляющейся и циклической структуры для решения задачи	При выполнении заданий обучающийся не смог обосновать использование алгоритмов линейной, разветвляющейся и циклической структуры для решения задачи	При выполнении заданий обучающийся с незначительными ошибками обосновал использование алгоритмов линейной, разветвляющейся и циклической структуры для решения задачи	При выполнении заданий обучающийся обосновал использование алгоритмов линейной, разветвляющейся и циклической структуры для решения задачи	При выполнении заданий обучающийся сравнил характеристики и самостоятельно обосновал использование алгоритмов линейной, разветвляющейся и циклической структуры для решения задачи
Умение использовать принципы работы с файловым типом данных при решении задачи	При выполнении заданий обучающийся не смог использовать принципы работы с файловым типом данных при решении задачи	При выполнении заданий обучающийся с незначительными ошибками использовал принципы работы с файловым типом данных при решении задачи	При выполнении заданий обучающийся использовал принципы работы с файловым типом данных при решении задачи	При выполнении заданий обучающийся самостоятельно сформулировал и использовал принципы работы с файловым типом данных при решении задачи
Умение соотнести полученный	При выполнении заданий	При выполнении заданий	При выполнении заданий	При выполнении заданий

результат с поставленной целью	обучающийся не смог соотнести полученный результат с поставленной целью	обучающийся смог с незначительными ошибками соотнести полученный результат с поставленной целью	обучающийся смог соотнести полученный результат с поставленной целью	обучающийся самостоятельно смог соотнести полученный результат с поставленной целью
--------------------------------	---	---	--	---

Оценка сформированности компетенций по показателю Навыки

Критерий	Уровень освоения и оценка			
	не зачтено	зачтено		
Практическая реализация алгоритмов линейной, разветвляющейся и циклической структуры с помощью интегрированной среды PyCharm	При выполнении заданий обучающийся не смог выполнить практическую реализацию алгоритмов линейной, разветвляющейся и циклической структуры с помощью среды PyCharm	При выполнении заданий обучающийся с незначительными ошибками смог выполнить практическую реализацию алгоритмов линейной, разветвляющейся и циклической структуры с помощью среды PyCharm	При выполнении заданий обучающийся с ошибками смог выполнить практическую реализацию алгоритмов линейной, разветвляющейся и циклической структуры с помощью среды PyCharm	При выполнении заданий обучающийся смог выбрать алгоритм решения задачи и выполнить практическую реализацию алгоритмов линейной, разветвляющейся и циклической структуры с помощью среды PyCharm
Применение файлов для организации ввода-вывода данных в среде PyCharm	При выполнении заданий обучающийся не смог применить файлы для организации ввода-вывода данных в среде PyCharm	При выполнении заданий обучающийся с незначительными ошибками смог применить файлы для организации ввода-вывода данных в среде PyCharm	При выполнении заданий обучающийся смог применить файлы для организации ввода-вывода данных в среде PyCharm	При выполнении заданий обучающийся смог выбрать требуемый тип файлов и применить файлы для организации ввода-вывода данных в среде PyCharm
Анализ результатов выполненных заданий	При выполнении заданий обучающийся не выполнил анализ результатов	При выполнении заданий обучающийся с незначительными ошибками выполнил анализ результатов	При выполнении заданий обучающийся выполнил анализ результатов	При выполнении заданий обучающийся самостоятельно выполнил анализ результатов

При промежуточной аттестации в форме дифференцированного зачета в семестре №3 используется следующая шкала оценивания:

2 – неудовлетворительно, 3 – удовлетворительно, 4 – хорошо, 5 – отлично.

Наименование показателя оценивания результата обучения по дисциплине	Критерий оценивания
ОПК-3. Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	

ОПК-3.2 Разрабатывает алгоритмы, выполняет отладку и тестирование программ с целью применения их в сфере информационно-коммуникационных технологий

Знания	Знание основ языка программирования Python, включая типы данных, управляющие структуры, операторы, стандартные библиотеки функций
	Объем освоенного материала
	Полнота ответов на вопросы
	Четкость изложения и интерпретации знаний
Умения	Полнота, качество выполненного задания
	Умение обосновать использование для решения задачи необходимых структурированных типов данных
	Умение использовать принципы отладки и тестирования программ для выявления синтаксических и логических ошибок
	Умение соотнести полученный результат с поставленной целью
Навыки	Применение массивов, строк, множеств, процедур и функций при написании программы решения задачи в среде PyCharm
	Применение программного инструментария для выполнения отладки и тестирования программного кода
	Анализ результатов выполненных заданий

Оценка сформированности компетенций по показателю Знания.

Критерий	Уровень освоения и оценка				
	2	3	4	5	
Знание основ языка программирования Python, включая типы данных, управляющие структуры, операторы, стандартные библиотеки функций	Не знает основ языка программирования Python, включая типы данных, управляющие структуры, операторы, стандартные библиотеки функций	Знает основы языка программирования Python, включая типы данных, управляющие структуры, операторы, стандартные библиотеки функций, но допускает неточности формулировок	Знает основы языка программирования Python, включая типы данных, управляющие структуры, операторы, стандартные библиотеки функций	Знает основы языка программирования Python, включая типы данных, управляющие структуры, операторы, стандартные библиотеки функций, может корректно сформулировать их самостоятельно	
Объем освоенного материала	Не знает значительной части материала дисциплины	Знает только основной материал дисциплины, не усвоил его деталей	Знает материал дисциплины в достаточном объеме	Обладает твердым и полным знанием материала дисциплины, владеет дополнительными знаниями	
Полнота ответов на вопросы	Не дает ответы на большинство вопросов	Дает неполные ответы на все вопросы	Дает ответы на вопросы, но не все - полные	Дает полные, развернутые ответы на поставленные вопросы	
Четкость изложения и интерпретации знаний	Излагает знания без логической последовательности. Неверно излагает и интерпретирует знания	Излагает знания с нарушениями в логической и последовательности. Допускает неточности в изложении и интерпретации знаний	Излагает знания без нарушений в логической и последовательности. Грамотно и по существу излагает знания	Излагает знания в логической последовательности, самостоятельно их интерпретируя и анализируя. Грамотно и точно излагает знания, делает самостоятельные выводы	

Оценка сформированности компетенций по показателю Умения.

Критерий	Уровень освоения и оценка			
	2	3	4	5
Полнота, качество выполненного задания	Задание не выполнено или выполнено некачественно	Задание выполнено с незначительными ошибками в полном объеме и качественно	Задание выполнено в полном объеме и качественно	Задание выполнено в полном объеме. Обучающимся сформулированы самостоятельные выводы, выполнен анализ полученных
Умение обосновать использование для решения задачи необходимых структурированных типов данных	При выполнении заданий обучающийся не смог обосновать использование для решения задачи необходимых структурированных типов данных	При выполнении заданий обучающийся с незначительными ошибками обосновал использование для решения задачи необходимых структурированных типов данных	При выполнении заданий обучающийся обосновал использование для решения задачи необходимых структурированных типов данных	При выполнении заданий обучающийся сравнил характеристики и самостоятельно обосновал использование для решения задачи необходимых структурированных типов данных
Умение использовать принципы отладки и тестирования программ для выявления синтаксических и логических ошибок	При выполнении заданий обучающийся не смог использовать принципы отладки и тестирования программ для выявления синтаксических и логических ошибок	При выполнении заданий обучающийся с незначительными ошибками использовал принципы отладки и тестирования программ для выявления синтаксических и логических ошибок	При выполнении заданий обучающийся использовал принципы отладки и тестирования программ для выявления синтаксических и логических ошибок	При выполнении заданий обучающийся самостоятельно сформулировал и использовал принципы отладки и тестирования программ для выявления синтаксических и логических ошибок
Умение соотнести полученный результат с поставленной целью	При выполнении заданий обучающийся не смог соотнести полученный результат с поставленной целью	При выполнении заданий обучающийся смог с незначительными ошибками соотнести полученный результат с поставленной целью	При выполнении заданий обучающийся смог соотнести полученный результат с поставленной целью	При выполнении заданий обучающийся самостоятельно смог соотнести полученный результат с поставленной целью

Оценка сформированности компетенций по показателю **Навыки**

Критерий	Уровень освоения и оценка			
	2	3	4	5
Применение массивов, строк, множеств, процедур и функций при написании программы решения задачи в среде PyCharm	При выполнении заданий обучающийся не смог применить массивы, строки, множества, процедуры и функции при написании программы решения задачи в среде PyCharm	При выполнении заданий обучающийся с незначительными ошибками смог применить массивы, строки, множества, процедуры и функции при написании программы решения задачи в среде PyCharm	При выполнении заданий обучающийся применил массивы, строки, множества, процедуры и функции при написании программы решения задачи в среде PyCharm	При выполнении заданий обучающийся смог самостоятельно определить тип данных и применить массивы, строки, множества, процедуры и функции при написании программы решения задачи в среде Python
Применение программного инструментария для выполнения отладки и тестирования программного кода	При выполнении заданий обучающийся не смог применить программный инструментарий для выполнения отладки и тестирования программного кода	При выполнении заданий обучающийся с незначительными ошибками смог применить программный инструментарий для выполнения отладки и тестирования программного кода	При выполнении заданий обучающийся применил программный инструментарий для выполнения отладки и тестирования программного кода	При выполнении заданий обучающийся смог составить тестовые данные и применил программный инструментарий для выполнения отладки и тестирования программного кода
Анализ результатов выполненных заданий	При выполнении заданий обучающийся не выполнил анализ результатов	При выполнении заданий обучающийся с незначительными ошибками выполнил анализ результатов	При выполнении заданий обучающийся выполнил анализ результатов	При выполнении заданий обучающийся самостоятельно выполнил анализ результатов

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

6.1. Материально-техническое обеспечение

№	Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы
1	Учебная аудитория для проведения лекционных занятий, консультаций	Специализированная мебель, мультимедийный проектор, переносной экран, ноутбук
2	Компьютерный класс для проведения практических занятий, текущего контроля, промежуточной аттестации	Специализированная мебель, компьютеры, обеспечивающие доступ к локальной сети университета и сети Интернет, переносной мультимедийный проектор, принтер
3	Читальный зал библиотеки для самостоятельной работы	Специализированная мебель; компьютерная техника, подключенная к сети «Интернет», имеющая доступ в электронную информационно-

№	Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы
		образовательную среду
4	Методический кабинет для самостоятельной работы	Специализированная мебель; мультимедийный проектор, переносной экран, ноутбук

6.2. Лицензионное и свободно распространяемое программное обеспечение

№	Перечень лицензионного программного обеспечения.	Реквизиты подтверждающего документа
1	Операционная система ASTRA LINUX Вариант лицензирования «Орел» 1.7	Контракт №144-22 от 27.10.2022 лицензия №223100026-alse-1.7-client-base_orel-x86_64-0-11874 от 07.11.2022 Лицензия бессрочная
2	Офисный пакет Мой офис Профессиональный 2.	Договор №143-22 от 31.10.2022 Лицензия бессрочная
3	Kaspersky Endpoint Security «Расширенный Russian Edition»	Контракт № 03261000041230000160001 «Поставка продления права пользования (лицензии) Kaspersky Endpoint Security от 21.08.2023. Срок действия лицензии 26.08.2025.
4	Yandex browser	Свободно распространяемое ПО согласно условиям лицензионного соглашения
5	Mozilla Firefox	Свободно распространяемое ПО согласно условиям лицензионного соглашения
6	Система компьютерного тестирования Online Test Pad	Свободно распространяемое ПО согласно условиям лицензионного соглашения

6.3. Перечень учебных изданий и учебно-методических материалов

1. Андреева, Т. А. Программирование на языке Python : учебное пособие / Т. А. Андреева. – 3-е изд. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. – 277 с. – ISBN 978-5-4497-0688-1. – URL: <https://www.iprbookshop.ru/97576.html>
2. Гавриловская, С.П. Программирование : лабораторный практикум : учебное пособие /С.П. Гавриловская. – Белгород: Изд-во БГТУ, 2017. – 87 с. – URL: <https://elib.bstu.ru/Reader/Book/2017101712393192400000657161>
3. Златопольский, Д. М. Программирование: типовые задачи, алгоритмы, методы / Д. М. Златопольский. – 4-е изд. – Москва : Лаборатория знаний, 2020. – 224 с. – ISBN 978-5-00101-789-9. – URL: <https://www.iprbookshop.ru/12264.html>
4. Роганов, Е. А. Основы информатики и программирования : учебное пособие / Е. А. Роганов. – 3-е изд. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. – 390 с. – ISBN 978-5-4497-0908-0. – URL: <https://www.iprbookshop.ru/102026.html>
5. Тюльпинова, Н. В. Технология алгоритмизации и программирования на языке Python : учебное пособие / Н. В. Тюльпинова. – Саратов : Вузовское образование, 2019. – 244 с. – ISBN 978-5-4487-0471-0. – URL: <https://www.iprbookshop.ru/80540.html>

6.4. Перечень интернет ресурсов, профессиональных баз данных, информационно-справочных систем

1. Электронно-библиотечная система: [сайт]. URL: <http://www.iprbookshop.ru>
2. Электронно-библиотечная система: [сайт]. URL: <https://e.lanbook.com/>
3. Научно-техническая библиотека БГТУ им. В.Г. Шухова: [сайт]. URL: <http://ntb.bstu.ru/>.
4. Справочно-правовая система КонсультантПлюс: [сайт]. URL: <http://www.consultant.ru/>.
5. Цифровая статистика: [сайт]. URL: https://www.economy.gov.ru/material/directions/gosudarstvennoe_upravlenie/cifrovaya_statistika/