

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
САМАРСКИЙ УНИВЕРСИТЕТ ГОСУДАРСТВЕННОГО УПРАВЛЕНИЯ
«МЕЖДУНАРОДНЫЙ ИНСТИТУТ РЫНКА»

УТВЕРЖДАЮ
Проректор по учебной работе и
качеству образования

_____ И. А. Долгова

15 апреля 2026 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ

ПРОГРАММИРОВАНИЕ

Направление подготовки:	09.03.03 Прикладная информатика
Профиль подготовки:	Корпоративные информационные системы
Квалификация:	бакалавр
Форма обучения:	очно-заочная
Год начала подготовки:	2026

Самара
2026

1. ОЦЕНОЧНЫЕ СРЕДСТВА, СООТНЕСЁННЫЕ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Код и наименование универсальной компетенции	Код и наименование индикатора достижения универсальной компетенции	Результаты обучения по дисциплинам и практикам	Вид аттестации и оценочных средств
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.И-1. Использует современные информационные технологии при решении задач профессиональной деятельности	ОПК-2.И-1.3-1. Знает тенденции развития современных информационных технологий	Текущий контроль: промежуточный тест, лабораторная работа. Промежуточная аттестация: вопросы на экзамене
		ОПК-2.И-1.У-1. Умеет использовать современные информационные технологии в профессиональной деятельности	Текущий контроль: промежуточный тест, лабораторная работа. Промежуточная аттестация: вопросы на экзамене
	ОПК-2.И-2. Использует современные программные средства, в том числе, отечественного производства при решении задач профессиональной деятельности	ОПК-2.И-2.3-1. Знает тенденции развития современных программных средств	Текущий контроль: промежуточный тест, лабораторная работа. Промежуточная аттестация: вопросы на экзамене
		ОПК-2.И-2.У-1. Умеет использовать современные программные средства в профессиональной деятельности	Текущий контроль: промежуточный тест, лабораторная работа, Промежуточная аттестация: вопросы на экзамене

2. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

2.1. Темы лабораторных работ

Раздел 1. Введение. Базовые понятия языка Python.

Лабораторная работа №1. Введение в программирование на языке Python.

Цель работы: Ознакомиться со средой разработки программ и базовыми элементами алгоритмического языка Python.

Вопросы для самопроверки:

1. Дайте расшифровку аббревиатуры IDE.
2. Какова структура кода программы на языке Python?
3. По каким правилам формируются имена переменных?
4. Какие виды циклов существуют?
5. В чем смысл применения функции `enumerate()`?

Раздел 2. Списки, матрицы, срезы.

Лабораторная работа №2. Базовая структура данных языка Python – списки.

Цель работы: Научиться работать со списками.

Вопросы для самопроверки:

1. Перечислите основные свойства списков.
2. Как можно список заполнить начальными значениями, в т.ч. и случайными?
3. Можно ли список заполнить данными разного типа?
4. Что такое список списков?
5. Как с помощью среза переписать элементы списка в обратном порядке?

Раздел 3. Функции и методы для списков.

Лабораторная работа №3. Стандартные функции для обработки списков.

Цель работы: Научиться использовать стандартные функции и методы для списков.

Вопросы для самопроверки:

1. Перечислите основные операции при обработке списков.
2. Можно ли складывать списки?
3. Чем операция добавления отличается от операции расширения списков?
4. Чем операция `del` отличается от функции `pop()`?
5. Список – это изменяемый или неизменяемый тип данных в Python?

Раздел 4. Работа с текстовыми строками.

Лабораторная работа №4. Стандартные функции для работы с текстовыми строками.

Цель работы: Научиться использовать стандартные функции и методы для работы с текстовыми строками.

Вопросы для самопроверки:

1. Строка – это изменяемый или неизменяемый тип данных в Python?
2. Можно ли складывать строки?
3. Чем строка отличается от списка символов?
4. Как из строки, состоящей из слов создать список из слов?
5. Что такое регулярные выражения?

Раздел 5. Кортежи и словари.

Лабораторная работа №5. Работа с кортежами и словарями.

Цель работы: Научиться использовать стандартные функции и методы для работы с кортежами и словарями.

Вопросы для самопроверки:

1. Кортеж – это изменяемый или неизменяемый тип данных в Python?
2. Словарь – это изменяемый или неизменяемый тип данных в Python?
3. Можно ли кортеж превратить в словарь, и, наоборот?
4. Чем ключ отличается от значения в словаре?
5. Можно ли сформировать список словарей?

Раздел 6. Функции в Python.

Лабораторная работа №6. Использование функций в программах на Python

Цель работы: Научиться создавать свои функции.

Вопросы для самопроверки:

1. Как объявить функцию в программе?
2. Можно ли определение функции записать после ее вызова?
3. Чем отличаются позиционные и именованные аргументы функции?
4. Что такое функция lambda?
5. В чем идея применения декоратора функции?

Раздел 7. Управление файловой системой.

Лабораторная работа №7. Программное управление файловой системой

Цель работы: Научиться программно манипулировать файлами и папками.

Вопросы для самопроверки:

1. Как создать папку?
2. Как удалить файл?
3. Как проверить существование файла?
4. Как вывести список файлов в заданной папке?
5. Как получить информацию о выбранном файле?

Раздел 8. Обработка текстовых файлов.

Лабораторная работа №8. Работа с текстовыми файлами.

Цель работы: Научится обрабатывать текстовые файлы.

Вопросы для самопроверки:

1. Дайте определение файла.
2. Чем текстовый файл отличается от бинарного?
3. Чем отличаются функции readline() и readlines()?
4. Для чего нужен оператор with при работе с файлами?
5. Зачем нужен формат JSON текстового файла?

Раздел 9. Множества.

Лабораторная работа №9. Работа со множествами.

Цель работы: Научится обрабатывать текстовые файлы.

Вопросы для самопроверки:

1. Множество – это изменяемый или неизменяемый тип данных?
2. Какое у множества базовое свойство?
3. Можно ли множество преобразовать в список?
4. Что такое пересечение множеств?
5. Что такое объединение множеств?

Цели лабораторных занятий:

1. Углубление и закрепление знания теоретического курса путем практического изучения в лабораторных условиях изложенных в лекциях методов и технологий;
2. Приобретение навыков в научном экспериментировании, анализе полученных результатов;
3. Формирование первичных навыков организации, планирования и проведения научных исследований.

Порядок проведения лабораторного занятия:

1. Вводная часть:
 - входной контроль подготовки обучаемого;
 - вводный инструктаж (знакомство обучающихся с содержанием предстоящей работы, краткий анализ теоретических положений и выводов, демонстрация подходов к выполнению отдельных операций, напоминание о технике безопасности, предупреждение о возможных ошибках).
2. Основная часть:
 - проведение обучаемым лабораторной работы;
 - текущее индивидуальное консультирование обучаемого;
3. Заключительная часть:
 - демонстрация результатов выполненного задания;
 - заключительный инструктаж (подведение итогов выполнения учебных задач, разбор допущенных ошибок и выявление их причин, сообщение результатов работы каждого обучаемого, объявление о том, что необходимо повторить к следующему занятию).

Особенности подготовки к проведению лабораторного занятия

Подготовка лабораторного занятия начинается с изучения теоретических положений, определения (уточнения) целей и задач данного занятия, времени, выделяемого обучаемым для подготовки.

В ходе подготовки к лабораторной работе необходимо пояснить проблематику, объем и содержание лабораторного занятия, определить, какие понятия, определения, теории могут быть иллюстрированы данным экспериментом, какие умения и навыки должны приобрести обучаемые в ходе занятия, какие знания углубить и расширить.

При этом преподавателю необходимо решить, на каком этапе обучения следует поставить задачу о подготовке к лабораторной работе, каким образом достигнуть активизации познавательной деятельности обучающихся. Задача на подготовку к лабораторной работе может быть поставлена на лекции, с таким временным расчетом, чтобы обучаемые смогли качественно подготовиться к ее проведению. Одновременно им выдаются учебно-методические материалы, иллюстрирующие круг вопросов, затрагиваемых в ходе выполнения лабораторного задания. Это могут быть методические указания по соответствующему курсу, презентации, ссылки на Интернет-источники и др. Эти материалы могут отражать учебные вопросы, краткие сведения по теории, программу выполнения работы, содержание отчета, вопросы для подготовки и литературу, рекомендуемую к изучению и т.д. В них также ставятся задачи, которые обучаемые должны решить при подготовке к работе, в процессе эксперимента и при обработке полученных результатов.

В ходе подготовки к лабораторной работе необходимо обратить внимание обучающегося на результат ее выполнения. Результат лабораторной работы должен быть четко сформулирован, приведены критерии его достижения, перечень материалов, его (результат) иллюстрирующих – файлы, графики, скриншоты и т.д. Учащийся должен уметь формулировать основные выводы, опираясь на полученный на лабораторной работе результат.

В отдельных случаях, на лабораторном занятии может быть предусмотрена защита выполненной работы.

Шкала и критерии оценки лабораторной работы

Критерии	Баллы
Степень соответствия выполненного задания поставленным требованиям	25
Структурирование и комментирование лабораторной работы	25
Уникальность выполненной работы (отличие от работ коллег)	25
Ответы на контрольные вопросы	25

Лабораторная работа оценивается по 100 балльной шкале, баллы переводятся в оценки успеваемости следующим образом:

90 – 100 баллов – «отлично»;

70 – 89 баллов – «хорошо»;

50 – 69 баллов – «удовлетворительно»;

менее 50 баллов – «неудовлетворительно».

3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

3.1. Банк контрольных заданий (с указанием компетенции)

1. Прочитайте текст и установите соответствие. (ОПК-2.1)

Установите соответствие между ключевыми словами (1-4) и типами данных в Python (А-Г):

1. list	А. Кортеж
2. set	Б.Словарь
3. tuple	В. Множество
4. dict	Г.Список

А	Б	В	Г

2. Прочитайте текст и установите последовательность (ОПК-2.1)

Установите правильную последовательность операторов для решения следующей задачи: ввести несколько слов, разделенных пробелами, вывести в консоль эти слова, отсортированные сначала по возрастанию, потом по убыванию

- А. **print(mbuf)**
- Б. **mbuf=buf.split()**
- В. **mbuf.sort(reverse=True)**
- Г. **buf=input()**
- Д. **mbuf.sort()**

Ответ

3. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ОПК-2.1)

Для отображения выборки из базы данных, лучше всего подходит следующая структура:

- А. Множество строк
- Б. Список строк
- В. Список словарей
- Г. Кортеж строк
- Д. Кортеж чисел

Ответ

Основание

4. Прочитайте текст и запишите развёрнутый обоснованный ответ. (ОПК-2.1)

В каждой строке многострочного текстового файла содержатся названия различных городов, разделенных запятыми, причем названия могут повторяться произвольное количество раз. Опишите (словесно) алгоритм по формированию файла с уникальными, неповторяющимися названиями городов, отсортированных по алфавиту.

5. Прочитайте текст и установите соответствие. (ОПК-2.1)

Установите соответствие между характеристиками (1-2) и типами данных в Python (А-Б):

1. Изменяемый тип данных	А. Целое число
2. Неизменяемый тип данных	Б. Булево значение

А	Б

6. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ОПК-2.1)

Какое значение получит переменная **a** после выполнения следующей программной конструкции:

```
a=0
for i in range(1,3):
    a+=i+2
print(a)
```

- 1) 0
- 2) 5
- 3) 7
- 4) 9
- 5) программа не работает из-за ошибки

Ответ
Основание

7. Прочитайте текст и установите соответствие. (ОПК-2.1)

Установите соответствие между характеристиками (1-2) и типами данных в Python (А-Б):

1. Изменяемый тип данных	А. Текстовая строка
2. Неизменяемый тип данных	Б. Список

А	Б

8. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ОПК-2.1)

Какой тип данных у переменной **double** в результате выполнения следующей программной структуры:

```
double=lambda x: x**2
print(type(double))
```

- 1) целое число
- 2) вещественное число
- 3) функция
- 4) список
- 5) программа не работает из-за ошибки

Ответ
Основание

9. Прочитайте текст и установите соответствие. (ОПК-2.2)

Установите соответствие между характеристиками (1-2) и типами данных в Python (А-Б):

1. Изменяемый тип данных	А. Множество
2. Неизменяемый тип данных	Б. Кортеж

А	Б

10. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ОПК-2.2)

Какой значение будет выведено в консоль в результате выполнения следующей программной конструкции:

```
str=("мама", "папа", "дядя")
str[3]="тетя"
buf=str[2]+str[3]
print(buf)
```

- 1) "папа"
- 2) "дядятетя"
- 3) "мама папа дядя тетя"
- 4) "дядя тетя"
- 5) программа не сработает из-за ошибки

Ответ
Основание

11. Прочитайте текст и установите соответствие. (ОПК-2.2)

Установите соответствие между характеристиками (1-2) и типами данных в Python (А-Б):

1. Изменяемый тип данных	А. Словарь
2. Неизменяемый тип данных	Б. Список

А	Б

12. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ОПК-2.2)

Какой значение будет выведено в консоль в результате выполнения следующей программной конструкции:

```
a=5
b=2

def mul():
    a+=1
    b+=2
    return(a*b)
print(mul())
```

- 1) 10
- 2) 18

- 3) 24
 4) 0
 5) программа не работает из-за ошибки

Ответ
Основание

13. Прочитайте текст и установите соответствие. (ОПК-2.2)

Установите соответствие между характеристиками (1-2) и функцией Python (А-Б):

1. Одно значение	А. lambda
2. Несколько значений	Б. map()

А	Б

14. Установите соответствие между утверждениями (1-2) и результатами выполнения кода (А-Б): (ОПК-2.2)

1. В консоли увидим два одинаковых значения	А. a=10 print(id(a)) a+=1 print(id(a))
2. В консоли увидим два разных значения	Б. a=10 print(id(a)) b=a print(id(b))

А	Б
---	---

15. Установите соответствие между характеристиками (1-2) и функцией Python (А-Б): (ОПК-2.2)

1. Одно значение	А. reduce()
2. Несколько значений	Б. filter()

А	Б
---	---

3.2. Ключи к контрольным заданиям

№ задания	Верный ответ
1	A3 B4 B2 Г1
2	ГБДАВА
3	В Выборка из базы данных представляет собой совокупность однородных записей, т.е. значений из полей выборки. Такой же вид имеет структура данных Python – список словарей.
4	Для решения этой задачи проще всего можно использовать следующий алгоритм: 1. Считываем файл построчно. 2. Слова из каждой строки добавляем в список строк 3. В итоге получаем большой список из слов с повторяющимися названиями городов 4. Преобразуем список из слов во множество, в результате чего избавляемся от повторов 5. Преобразуем множество обратно в список и сортируем его по возрастанию 6. Записываем слова из этого списка в новый файл
5	A2B2
6	3 В функции range() не итерируется последнее значение диапазона
7	A2B1
8	3 Специальная функция lambda возвращает функцию
9	A1B2
10	5 Кортеж относится к неизменяемому типу данных
11	A1B1
12	5 a, b – глобальные переменные
13	A1B2
14	A2B1
15	A1B2

Шкала и критерии оценки текущего тестирования

Число правильных ответов	Оценка
90-100% правильных ответов	Оценка «отлично»
70-89% правильных ответов	Оценка «хорошо»
50-69% правильных ответов	Оценка «удовлетворительно»
Менее 50% правильных ответов	Оценка «неудовлетворительно»

3.3. Перечень тем для проверки образовательных результатов на знания (вопросы к экзамену)

1. Структура программы на Python, размещение операторов. Комментарии, переменные, типы данных, ключевые слова.
2. Базовые арифметические операции и встроенные функции. Логические операторы.
3. Циклы (while и for) – особенности реализации.
4. Функция range() и ее использование в цикле for. Оператор pass, оператор множественного выбора, функция enumerate().
5. Общие сведения о списках, их фундаментальные свойства.
6. Сравнение с массивами в других языках программирования.
7. Базовые алгоритмы обработки списков.
8. Инициализация списков, в т.ч. случайными значениями.
9. Использование отрицательных индексов.
10. Понятие матрицы (двумерного списка).
11. Базовые алгоритмы обработки матриц и их инициализация.
12. Понятие среза списка, примеры использования.
13. Базовые ошибки при работе со списками и матрицами, использование конструкции try-except-else-finally.
14. Операция "+" для списков. Сравнение списков.
15. Добавление элементов в существующий список (функция append).
16. Расширение списка (функция extend).
17. Методы count(), insert(), remove().
18. Методы pop(), clear(), copy().
19. Ключевое слово del и его отличие от метода pop().
20. Поиск индекса элемента списка по его значению.
21. Формирование текстовых переменных.
22. "Длинные" строки в Python.
23. Базовые операции со строками – конкатенация (сложение) и дублирование ("умножение"), f и r –форматирование строки.
24. Сравнение строк и списка символов – сходства и различия.
25. Работа со структурой "список строк".
26. Базовые операции со строками: выделение подстроки, замена, удаление и др.
27. Основные функции для работы со строками: поиск индекса вхождения подстроки в строку, очистка, преобразование символов, определение типа символов, центрирование, определение длины строки, разбиение, удаление и т.д.
28. Регулярные выражения – примеры использования.
29. Понятие кортежа.
30. Сравнение кортежа со списком.
31. Доступ к элементам кортежа.
32. Преобразование кортежа в список и обратно.
33. Понятие о словарях. Создание словаря.
34. Добавление и удаление объекта в словарях.
35. Базовые операции со словарями: перебор элементов, создание копии, получение значения поля по ключу, очистка словаря.
36. Получение ключей словаря -функция keys().
37. Получение значений словаря - функция values().
38. Получение пар ключ-значение - функция items().
39. Удаление элементов словаря, работа с не существующими ключами (setdefault), обновление словаря. Примеры работы со списком словарей.
40. Создание функции и ее назначение.
41. Примеры реализации простейших функций.
42. Функция, возвращающая кортеж и список.
43. Множественное описание функции. Многострочный комментарий в функции.

44. Формальные и фактические параметры в функциях Python.
45. Изменяемые и неизменяемые типы данных. Особенности их использования при передаче параметров в функцию.
46. Позиционные и именованные аргументы.
47. Функция с произвольным количеством позиционных и именованных аргументов.
48. Понятие о функциональном стиле программирования в Python. "Особые" функции Python: yield, lambda, map, reduce, filter.
49. Области видимости переменных в функциях.
50. Вложенные функции. Понятие о замыканиях и декораторах.
51. Работа с модулем os. Базовые операции модуля: получение текущей директории, создание папки, смена текущей директории, переименование, перемещение, удаление файлов.
52. Рекурсивный переход ко всем вложенным папкам.
- 53.** Получение информации о файле. Относительные пути из текущей директории.
54. Бинарные и текстовые файлы. Вывод текстового файла с автоматической нумерацией строк.
55. Считывание файла в список строк.
56. Запись списка строк в новый файл.
57. Режимы чтения и записи файлов.
58. Использование оператора with при работе с файлами.
59. Различные способы считывания текстовых файлов: read(), readline(), readlines().
60. Функции записи информации в текстовый файл. Примеры на работу с текстовыми файлами.
61. Пользовательские модули и пакеты.
62. Работа с файлами разных форматов: CSV, XML, JSON, EXCEL.
63. Особенности типа данных "множество". Создание множества.
64. Преобразование множества в кортеж, список, словарь.
65. Основные операции над множествами: перебор элементов, генератор множества, добавление элементов, удаление элементов.
66. Пример программы на удаление повторяющихся слов в текстовом файле.
67. Действия со множествами: пересечение, объединение, вычитание, симметричная разность, сравнение.
68. Понятие о комплексных числах в Python.