Приложение 4
к образовательной программе

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.01.02.02 Информатика и программирование
(индекс, наименование дисциплины в соответствии с учебным планом)

09.03.03 Прикладная информатика
(код, наименование направления подготовки/специальности)

Прикладная информатика в управлении корпоративными информационными системами
(наименование образовательной программы)

Очная форма обучения
(форма обучения)

Год набора – 2026
Донецк

Автор(ы)-составитель(и) РПД:

Лебезова Элла Михайловна, старший преподаватель кафедры информационных технологий

Заведующий кафедрой:

Брадул Наталья Валерьевна, канд. физ.-мат. наук, заведующий кафедрой информационных технологий

Рабочая программа дисциплины Б1.О.01.02.02 Информатика и программирование одобрена на заседании кафедры информационных технологий администрирования факультета государственной службы и управления Донецкого филиала РАНХиГС.

Протокол № 7 от «05 » марта 2026 г.

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Дисциплина Б1.О.01.02.02 Информатика и программирование обеспечивает формирование у обучающихся следующих общепрофессиональных компетенций*:

ОТФ /ТФ и реквизиты ПС *(при наличии)* **	**Код компетенции ****	**Наименование Компетенции ****	**Код индикатора достижения компетенций ****	**Наименование индикатора достижения компетенций ****	**Образовательный результат ****
-	ОПК-7.	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.3.	Разрабатывает алгоритмы и программы, пригодные для практического применения	**Знает** базовые алгоритмы и структуры данных. **Умеет** разрабатывать алгоритмы и программы для решения типовых задач.

** Дисциплина может формировать компетенцию полностью или частично.*
***Должно соответствовать Приложению 1 к образовательной программе*

2. Объем и место дисциплины в структуре образовательной программы

Общий объем дисциплины:

3,00 з.е., 108 ак.час

Контактная работа обучающихся с преподавателем по видам учебных занятий: 57 ак. час на контактную работу с преподавателем, из них 24 ак. час на лекции и 24 ак. час на практические занятия. 51 ак. час на самостоятельную работу обучающихся.

Б1.О.01.02.02. Информатика и программирование реализуется в 1-м семестре 1-го курса.

.

3. Содержание и структура дисциплины

3.1. Структура дисциплины

Очная форма обучения

№ п/п	Наименование тем и (или) разделов	ВСЕГО	Объем дисциплины, ак.час												Форма текущего контроля успеваемости, промежуточной аттестации
			Контактная работа обучающихся с преподавателем по видам учебных занятий									Самостоятельная работа			
			Период теоретического обучения						Период промежуточной аттестации (сессия)						
			Занятия лекционного типа		Занятия семинарского типа		ИК	КСР	КЭ	Каттэк	Контроль	СРкр	СРэк	СР	
			Л	ВЛ	ЛР	ПЗ									
РАЗДЕЛ 1. Введение в алгоритмизацию на Python															
Тема 1	Введение в программирование, как раздела информатики. Язык программирования Python.	11	2	0	0	2	0	0	0	0	0	0	0	7	Контрольные вопросы, практические занятия, КР1
Тема 2	Ввод-вывод, базовые типы	15	4	0	0	4	0	0	0	0	0	0	0	7	Контрольные вопросы,

	данных, основные операции Python														практические занятия, КР1
Тема 3	Ветвления в программах с помощью операторов принятия решений	15	4	0	0	4	0	0	0	0	0	0	0	7	Контрольные вопросы, практические занятия, КР 1
Тема 4	Циклические алгоритмы, операторы циклов Python	15	2	0	0	2	0	0	0	0	0	0	0	7	Контрольные вопросы, практические занятия, КР 1
РАЗДЕЛ 2. Использование строк и списков.															
Тема 5	Строки в Python. Функции и методы работы со строками	15	4	0	0	4	0	0	0	0	0	0	0	7	Контрольные вопросы, практические занятия, КР 2
Тема 6	Списки. Списочные выражения.	16	4	0	0	4	0	0	0	0	0	0	0	8	Контрольные вопросы, практические занятия, КР 2
Тема 7	Генерация случайных чисел в программировании задач моделирования и игр.	12	2	0	0	2	0	0	0	0	0	0	0	8	Контрольные вопросы, практические занятия, КР 2

Промежуточная аттестация	9	0	0	0	0	0	0	0	9	0	0	0	0	Зачет с оценкой
Итого	**108**	**24**	**0**	**0**	**24**	**0**	**0**	**0**	**9**	**0**	**0**	**0**	**51**	

Используемые сокращения:

Л – лекции - занятия, предусматривающие преимущественную передачу учебной информации обучающимся педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях,).

ВЛ – видео лекции.

ЛР – лабораторные работы.

ПЗ – практические занятия (за исключением лабораторных работ).

ИК – индивидуальные консультации.

КСР – контроль самостоятельной работы

КЭ – консультации перед экзаменом

Каттэк – контактная работа на аттестацию в период экзаменационных сессий

Контроль - контактная работа на аттестацию в период экзаменационных сессий для заочной формы обучения

СРкр – самостоятельная работа на подготовку курсовой работы/ курсового проекта.

СРэк – самостоятельная работа на подготовку к экзамену.

СР – самостоятельная работа в семестре на подготовку к учебным занятиям.

3.2. Содержание дисциплины

РАЗДЕЛ 1. ВВЕДЕНИЕ В АЛГОРИТМИЗАЦИЮ НА PYTHON

Тема 1. Введение в программирование, как раздела информатики. Язык программирования Python. ОПК-7.3.

Программирование как научная и прикладная дисциплина. Эволюция языков программирования. Язык Python: история создания, области применения (научные вычисления, веб-разработка, автоматизация, обработка данных). Интерпретируемость и динамическая типизация. Установка и настройка среды разработки. Структура простейшей программы. Понятие алгоритма: свойства (дискретность, детерминированность, результативность, массовость). Способы описания алгоритмов (словесный, графический, псевдокод, программный).

Тема 2. Ввод-вывод, базовые типы данных, основные операции Python. ОПК-7.3.

Функции вывода данных на экран и ввода данных с клавиатуры, форматирование вывода (строки с подстановкой значений). Базовые типы: целые числа, вещественные числа, логический тип, строки, пустой тип. Преобразование типов (явное и неявное). Арифметические операции: сложение, вычитание, умножение, деление, целочисленное деление, остаток от деления, возведение в степень. Операции сравнения: равно, не равно, меньше, больше, меньше или равно, больше или равно. Логические операции: «and», «or», «not». Приоритет выполнения операций. Понятие переменной, правила именования. Комментарии в коде.

Тема 3. Ветвления в программах с помощью операторов принятия решений. ОПК-7.3.

Логический тип данных и условные выражения. Условный оператор «if». Блок «else». Каскадное ветвление «elif». Вложенные условные конструкции. Тернарный условный оператор (условное выражение в одну строку). Проверка принадлежности значения диапазону. Обработка множественных условий. Типовые задачи: определение максимума и минимума, решение квадратного уравнения, классификация вводимых данных, реализация простейшего калькулятора с выбором действия.

Тема 4. Циклические алгоритмы, операторы циклов Python. ОПК-7.3.

Понятие цикла: тело цикла, условие продолжения, счетчик. Цикл «while»: синтаксис, применение при неизвестном числе итераций. Цикл «for»: итерация по диапазону, по символам строки, по элементам последовательности. Управление циклом: досрочный выход, переход к следующей итерации. Бесконечные циклы и способы их предотвращения. Вложенные циклы. Типовые задачи: вычисление факториала, суммы ряда, табуляция функции, вывод числовых паттернов.

РАЗДЕЛ 2. ИСПОЛЬЗОВАНИЕ СТРОК И СПИСКОВ

Тема 5. Строки в Python. Функции и методы работы со строками. ОПК-7.3.

Строка как неизменяемая последовательность символов. Индексация (прямая и обратная). Операция среза (извлечение подстроки по начальному, конечному индексу и шагу). Конкатенация (объединение) и повторение строк. Основные методы: преобразование в верхний и нижний регистр, преобразование первого символа в заглавный, удаление пробелов по краям, разбиение строки на части по разделителю, склеивание списка строк в одну, поиск подстроки, замена одной подстроки на другую, подсчет вхождений. Проверка свойств строки: состоит ли строка только из цифр, только из букв, только из букв и цифр, состоит ли только из пробельных символов. Форматирование строк. Сравнение строк. Типовые задачи: анализ текста, очистка ввода, извлечение данных из строки.

Тема 6. Списки. Списочные выражения. ОПК-7.3.

Список как изменяемая упорядоченная коллекция. Создание списков: перечислением элементов в квадратных скобках, с помощью функции создания списка, генераторы списков (компактная запись для заполнения списка по правилу). Операции со списками: индексация, срезы, конкатенация, повторение. Основные методы: добавление элемента в конец, расширение списка другим списком, вставка элемента по индексу, удаление элемента по значению, удаление и возврат элемента по индексу, поиск индекса элемента, подсчет вхождений элемента, сортировка по возрастанию, разворот списка. Встроенные функции получения длины, суммы, минимального и максимального элемента. Вложенные списки (матрицы). Копирование списков (поверхностное и глубокое). Итерация по элементам с помощью цикла «for».

Тема 7. Генерация случайных чисел в программировании задач моделирования и игр. ОПК-7.3.

Понятие псевдослучайных чисел. Модуль для работы со случайными величинами. Функции: получение случайного вещественного числа от нуля до единицы, получение случайного целого числа в заданном диапазоне, получение случайного целого числа с указанным шагом, получение случайного вещественного числа в диапазоне, случайный выбор элемента из последовательности, случайное перемешивание элементов последовательности, получение случайной выборки заданного размера. Инициализация генератора с помощью начального значения (для воспроизводимости результатов). Типовые задачи: угадай число, генерация пароля, симуляция бросков монеты.

4. Типы оценочных материалов, показатели и критерии оценивания

4.1. Оценочные материалы по дисциплине Б1.О.01.02.02. Информатика и программирование входят в состав оценочных материалов по образовательной программе. Совокупность оценочных материалов по всем дисциплинам (модулям) образовательной программы составляет фонд оценочных средств (далее – ФОС). ФОС используется при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся с целью оценивания достижения обучающимися планируемых результатов обучения.

4.2. ФОС разработан как комплекс проверочных заданий различного типа и уровня сложности, включает критерии и шкалы оценивания, а также «ключи» правильных ответов. ФОС формируется как отдельный документ и хранится в электронном виде, доступ к ФОС предоставлен ограниченному кругу лиц.

4.3. Для самостоятельной работы обучающихся при подготовке к текущему контролю успеваемости и промежуточной аттестации в рабочих программах дисциплин размещены типовые проверочные задания, которые можно условно разделить на задания закрытого, комбинированного и открытого типов.

Задания закрытого типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных.

Задания комбинированного типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных и обосновать свой выбор.

Задания открытого типа – это задания, в которых на каждый вопрос должен быть предложен развернутый обоснованный ответ.

В зависимости от типа задания рекомендованы определенная последовательность выполнения и система оценивания выполнения заданий.

4.4. Типы заданий, сценарии выполнения, критерии оценивания

ТИП ЗАДАНИЯ	ИНСТРУКЦИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	КРИТЕРИИ ОЦЕНИВАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных	Прочитайте текст, выберите правильный ответ	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В).	Ответ считается верным, если правильно указана цифра или буква
Задание закрытого типа на установление соответствия	Прочитайте текст и установите соответствие	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).	Ответ считается верным, если правильно указаны цифры или буквы

Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных	Прочитайте текст, выберите правильные ответы	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).	Ответ считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)
Задание закрытого типа на установление последовательности	Прочитайте текст и установите последовательность	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты ответа. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135).	Ответ считается верным, если правильно указана вся последовательность цифр

Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора	Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа. 5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования).	Ответ считается верным, если правильно указана цифра или буква и приведены корректные аргументы, используемые при выборе ответа
Задание открытого типа с развернутым ответом	Прочитайте текст и запишите развернутый обоснованный ответ	1. Внимательно прочитать текст задания и понять суть вопроса. 2. Продумать логику и полноту ответа. 3. Записать ответ, используя четкие компактные формулировки. 4. В случае расчетной задачи, записать решение и ответ	Ответ считается верным: 1. Отсутствие фактических ошибок. 2. Раскрытие объема используемых понятий (полнота ответа). 3. Обоснованность ответа (наличие аргументов). 4. Логическая последовательность излагаемого материала.

4.5. Общая шкала оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся с применением БРС

Оценка по шкале ECTS	**Сумма баллов за все виды учебной деятельности**	**Оценка по государственной шкале**	**Определение**
A	90 – 100	«Отлично»	отличное выполнение с незначительным количеством неточностей
B	80 – 89	«Хорошо»	в целом правильно выполненная работа с незначительным количеством ошибок (до 10%)
C	75 – 79		в целом правильно выполненная работа с незначительным количеством ошибок (до 15%)
D	70 – 74	«Удовлетворительно»	неплохо, но со значительным количеством недостатков
E	60 – 69		выполнение удовлетворяет минимальные критерии
FX	35 – 59	«Не удовлетворительно»	с возможностью повторной сдачи
F	0 – 34		с обязательным повторным изучением дисциплины (выставляется комиссией)

Соотношение баллов за текущий контроль успеваемости и промежуточную аттестацию, а также повторную промежуточную аттестацию:

Максимальная сумма баллов за текущий контроль успеваемости	Максимальная сумма баллов за промежуточную аттестацию	Максимальная итоговая балльная оценка	Максимальная сумма баллов за повторную промежуточную аттестацию
100 баллов	100 баллов	100 баллов	100 баллов

5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам

5.1. В ходе реализации дисциплины Б1.О.01.02.02 Информатика и программирование используются следующие формы текущего контроля успеваемости обучающихся (в том числе, задания к контрольным точкам):

Контрольные вопросы для проведения опроса, задания открытого типа на практических занятиях, контрольные задания

Таблица 5.1.

Распределение баллов по видам учебной деятельности (БРС)

Раздел/Темы	Формы текущего контроля		КЗР
	УО	ПЗ	
Р-1. / Т-1	3	7	15
Р-1. / Т-2	3	7	
Р-1. / Т-3	3	7	
Р-1. / Т-4	3	7	15
Р-2. / Т-5	3	7	
Р-2. / Т-6	3	7	
Р-2. / Т-7	3	7	
Итого: 100 б	18	7	30

УО – устный опрос;
ТЗ – тестовое задание;
КЗ – контрольные задания;
ПЗ – практическое занятие;
Д – доклад;
КЗР – контрольные работы по разделу.

Критерии оценивания опроса:

Балы	Описание критерия
3	Обучающийся полно излагает материал (отвечает на вопрос), дает правильное определение основных понятий; обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; излагает материал последовательно и правильно с точки зрения норм литературного языка.
2	Обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.
1	Обучающийся обнаруживает знание и понимание основных положений данной темы, но излагает материал неполно и допускает неточности в определении понятий или формулировке правил; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
0	Обучающийся обнаруживает незнание вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал.

0* - в журнал академической группы не выставляется

Критерии оценивания практических занятий:

Балы	Описание критерия	
3	Свыше 90% правильных ответов.	Обучающийся демонстрирует глубокое познание в освоенном материале.
2	Свыше 70% правильных ответов.	Обучающимся материал освоен полностью, без существенных ошибок.
1	Реализовано более 50% поставленных задач	Обучающимся материал освоен не полностью, имеются значительные пробелы в знаниях.
0	Реализовано менее 30% поставленных задач.	Обучающимся материал не освоен, знания обучающегося ниже базового уровня.

0* - в журнал академической группы не выставляется

Критерии оценивания контрольных заданий:

Балы	Описание критерия
12-15	Обучающимся задание выполнено без ошибок и в полном объеме.
8-11	Обучающимся в целом задание выполнено, имеются отдельные неточности или недостаточно полные ответы, не содержащие ошибок.
5-7	Обучающимся допущены отдельные ошибки при выполнении задания
0-4	У обучающегося отсутствуют ответы на большинство вопросов задачи, задание не выполнено или выполнено не верно.

0* - в журнал академической группы не выставляется

5.2. Типовые оценочные материалы для текущего контроля успеваемости обучающихся (вне контрольных работ):

РАЗДЕЛ 1. ВВЕДЕНИЕ В АЛГОРИТМИЗАЦИЮ НА PYTHON

Тема 1. Введение в программирование, как раздела информатики. Язык программирования Python.

Контрольные вопросы:

1. Почему программирование считается не просто набором технических навыков, а полноценным разделом информатики, и какие научные проблемы оно решает наряду с прикладными задачами?
2. Каковы основные этапы эволюции языков программирования от машинных кодов до современных высокоуровневых языков, и какое место в этой эволюции занимает Python?
3. Назовите не менее трёх различных областей практического применения языка Python и поясните, какие свойства языка делают его востребованным именно в этих сферах.
4. В чём заключается принципиальная разница между интерпретируемыми и компилируемыми языками, и к какой из этих групп относится Python?
5. Что означает понятие «динамическая типизация» в контексте языка Python, и какие преимущества и недостатки она даёт разработчику по сравнению со статической типизацией?

6. Перечислите основные способы описания алгоритмов (не менее четырёх) и кратко охарактеризуйте каждый из них, указав, в каких ситуациях какой способ предпочтительнее.

7. Назовите все пять свойств алгоритма согласно классическому определению (дискретность, детерминированность и так далее) и раскройте смысл каждого свойства на конкретном примере.

8. Какие компоненты входят в минимальную структуру простейшей программы на Python, и что происходит на каждом этапе — от написания исходного кода до получения результата?

Практические занятия:

Задание 1. Установите на свой компьютер интерпретатор Python (последней стабильной версии) и любую среду разработки. После установки создайте новый файл, напишите в нём единственную строку кода, которая выводит на экран фразу «Программирование – раздел информатики». Запустите программу и убедитесь, что вывод соответствует ожидаемому. Сохраните файл с именем introduction.py. В ответе на задание укажите путь к сохранённому файлу и сделайте скриншот окна с результатом выполнения.

Задание 2. Намеренно напишите программу с синтаксической ошибкой: пропустите кавычку при записи строки для вывода (например, напишите print(Привет) без кавычек). Запустите программу, внимательно прочитайте сообщение об ошибке, которое выдаст интерпретатор. Определите:

- номер строки, в которой обнаружена ошибка;
- тип ошибки (по первому слову в сообщении);
- предположительную причину возникновения.

Исправьте ошибку, добейтесь корректного выполнения программы. Запишите исходное сообщение об ошибке и его интерпретацию (что именно пошло не так).

Задание 3. Дан следующий фрагмент текста: «Взять число. Прибавить к нему 2. Если результат больше 10, то вычесть 5. Повторить сначала». Проанализируйте этот фрагмент на соответствие пяти свойствам алгоритма (дискретность, детерминированность, результативность, массовость, понятность). Укажите, каким свойствам данный набор инструкций удовлетворяет, а каким – нет, и почему. Если какое-либо свойство нарушено, предложите способ исправления.

Задание 4. Опишите алгоритм перехода через автомобильную дорогу с двусторонним движением по нерегулируемому пешеходному переходу словесно (пошагово, на естественном языке). Затем нарисуйте его блок-схему, используя стандартные графические элементы (начало/конец – овал, действие – прямоугольник, решение – ромб, стрелки переходов). Сравните оба способа: какой из них, на ваш взгляд, более нагляден и почему?

Задание 5. Создайте новый файл с именем about_algorithms.py. Напишите в нём программу, которая выводит на экран три строки:

- первая строка: название любого свойства алгоритма (например, «Дискретность»);

- вторая строка: краткое определение этого свойства своими словами;
- третья строка: пример алгоритма (самого простого, из 2–3 шагов), в котором это свойство проявляется.

Перед каждым из трёх блоков (вывод названия свойства, вывод определения, вывод примера) добавьте комментарий на русском языке, поясняющий, что делает следующая строка кода. Запустите программу, убедитесь в корректности вывода.

Задание 6. Используя любые доступные источники, составьте таблицу из трёх столбцов:

- «Язык программирования» (5–7 языков на выбор);
- «Уровень языка» (низкий, высокий, очень высокий);
- «Типизация» (статическая / динамическая);
- «Основная сфера применения» (1–2 предложения).

В таблицу обязательно включите Python, а также хотя бы один язык из каждой группы: машинно-ориентированный (ассемблер), классический компилируемый (C, C++, Java), узкоспециализированный (SQL, R, MATLAB). Результат оформите в текстовом виде или скриншоте.

Задание 7. Ниже приведён алгоритм, который претендует на решение задачи «определить максимальное из двух введённых чисел». Однако он не удовлетворяет свойству детерминированности: «Запросить у пользователя первое число. Запросить у пользователя второе число. Сравнить их. Вывести "Первое число больше или равно", если условие выполняется. В противном случае с вероятностью 50% вывести "Второе число больше", а с вероятностью 50% вывести "Первое число меньше"».

Ответьте на вопросы:

1. Почему данный алгоритм нельзя считать детерминированным?
2. В каких реальных системах (например, в игровых или моделирующих) подобная недетерминированность допустима или даже желательна?

Перепишите алгоритм так, чтобы он стал детерминированным.

Тема 2. Подготовка данных для машинного обучения

Контрольные вопросы:

1. Каким образом в Python организован вывод данных на экран и ввод данных с клавиатуры, и в чём заключается ключевое различие между этими двумя механизмами с точки зрения потока данных?
2. Что такое форматирование вывода, какие существуют способы подстановки значений в выводимую строку и какой из них считается современным и предпочтительным?
3. Перечислите все базовые типы данных, встроенные в Python (целые числа, вещественные числа, логический тип, строки, пустой тип), и приведите для каждого пример литерального представления.
4. В чём заключается разница между явным и неявным преобразованием типов, и в каких ситуациях Python выполняет

преобразование автоматически, а когда требуется вмешательство программиста?

5. Назовите все арифметические операции, доступные в Python, поясните, чем отличается обычное деление от целочисленного, и что возвращает операция нахождения остатка.

6. Какие операции сравнения существуют в Python, и какое значение (какого типа) возвращает каждая из них при вычислении?

7. Что такое логические операции «и», «или», «не», как они работают в соответствии с правилами алгебры логики, и каков приоритет этих операций по отношению друг к другу и к операциям сравнения?

8. Каким правилам должно подчиняться имя переменной в Python, какие имена являются допустимыми, а какие запрещёнными, и что такое «осмысленные имена» с точки зрения хорошего стиля программирования?

Практические занятия:

Задание 1. В популярном сериале «Остаться в живых» использовалась последовательность чисел 4 8 15 16 23 42, которая принесла героям удачу и помогла сорвать джекпот в лотерее. Напишите программу, которая выводит данную последовательность чисел с одним пробелом между ними.

Примечание. Текст '4 8 15 16 23 42' не использовать.

Задание 2. Измените предыдущую программу так, чтобы каждое число последовательности 4 8 15 16 23 42 печаталось на отдельной строке.

Задание 3. На вход программе подается строка текста – название футбольной команды. Напишите программу, которая повторяет ее на экране со словами « - чемпион!»

Задание 4. Напишите программу, которая выводит следующий треугольник, состоящий из звёздочек (*):

```
*
**
***
****
*****
******
*******
```

Рис. 1. Звёздный треугольник

Задание 5. Напишите программу, которая считывает три строки по очереди, а затем выводит их в обратной последовательности, каждую на отдельной строчке.

Примечание. По умолчанию команда print() принимает несколько аргументов (параметров), выводит их через один пробел, после чего ставит перевод строки. Это поведение можно изменить, используя необязательные именованные параметры sep (separator, разделитель) и end (окончание). Значения по умолчанию у параметров sep и end следующие:

sep=' '

end='\n'

Задание 6. Напишите программу, которая выводит на экран текст «I===like===Python» (без кавычек). Используйте необязательный параметр sep.

Задание 7. Напишите программу, которая считывает строку-разделитель и три строки с любым текстом, а затем выводит указанные строки через разделитель.

Задание 8. Напишите программу, которая выводит прямоугольник, по периметру состоящий из звездочек (*).

```
*****************
*               *
*               *
*****************
```

Рис. 2. Звездный прямоугольник

Примечание. Высота и ширина прямоугольника равны 4 и 17 звёздочкам соответственно.

Задание 9. Написать программу, которая, запрашивает два целых числа, и выводит на экран сумму и разность данных чисел.

Задание 10. Напишите программу, которая считывает два целых числа и выводит на экран квадрат суммы и сумму квадратов этих чисел. Пример вывода:

Квадрат суммы 3 и 2 равен 25

Сумма квадратов 3 и 2 равна 13

Задание 11. Напишите программу, которая приветствует пользователя, выводя слово «Привет» (без кавычек), после которого должна стоять запятая и пробел, а затем введенное имя и восклицательный знак.

Примечание. Перед восклицательным знаком не должно быть пробелов.

Задание 12. Напишите программу, которая запрашивает у пользователя целое число и выводит его квадрат и куб.

Задание 13. Напишите программу вывода на экран трех последовательно идущих чисел, каждое на отдельной строке. Первое число вводит пользователь, остальные числа вычисляются в программе.

Задание 14. Напишите программу, вычисляющую объём куба и площадь его полной поверхности по введённому значению длины ребра.

Задание 15. Напишите программу, которая считывает целое число, после чего на экран выводится следующее и предыдущее целое число с пояснительным текстом.

Пример при вводе 20:

Следующее за числом 20 число: 21

Для числа 20 предыдущее число: 19

Задание 16. Напишите программу, которая считает стоимость трех компьютеров, состоящих из монитора, системного блока, клавиатуры и

мыши. На вход программе подаётся четыре целых числа, каждое на отдельной строке. В первой строке – стоимость монитора, во второй строке – стоимость системного блока, в третьей строке – стоимость клавиатуры и в четвертой строке – стоимость мыши.

Программа должна вывести одно число – стоимость покупки (трех компьютеров).

Задание 17. Напишите программу, которая считывает целое положительное число x и выводит на экран последовательность чисел x, 2x, 3x, 4x и 5x, разделённых тремя черточками.

Пример вывода:

7---14---21---28---35

Задание 18. Напишите программу, которая находит полное число метров по заданному числу сантиметров.

Задание 19. Напишите программу для пересчёта величины временного интервала, заданного в секундах, в величину, выраженную в часах, минутах и секундах.

Задание 20. Напишите программу получения цифр трёхзначного числа:

Задание 21. Напишите программу, в которой рассчитывается сумма цифр двузначного числа.

Задание 22. Напишите программу, которая печатает число, образованное при перестановке цифр двузначного числа

Задание 23. Напишите программу, в которой рассчитывается сумма и произведение цифр положительного трёхзначного числа

Задание 24. Дано трехзначное число abc, в котором все цифры различны. Напишите программу, которая выводит шесть чисел, образованных при перестановке цифр заданного числа. Программа должна вывести шесть чисел, образованных при перестановке цифр заданного числа в следующем порядке: abc, acb, bac, bca, cab, cba.

Тема 3. Ветвления в программах с помощью операторов принятия решений

Контрольные вопросы:

1. Что такое условный оператор «if» в программировании, какова его общая структура, и при каких условиях выполняется тело этого оператора?
2. Для чего нужен блок «else» в конструкции ветвления, и чем отличается ситуация, когда этот блок присутствует, от ситуации, когда он отсутствует?
3. Что представляет собой каскадное ветвление с использованием «elif», и в каких случаях оно оказывается более удобным и читаемым, чем множество вложенных операторов «if»?
4. Объясните понятие «вложенные условные конструкции», приведите пример задачи, где без вложенных условий обойтись невозможно или крайне неудобно.
5. Что такое тернарный условный оператор, как он записывается, и в каких ситуациях его применение оправдано с точки зрения читаемости кода?

6. Как в Python выполняется проверка принадлежности значения заданному диапазону, и какие способы записи такого условия существуют?

7. Как правильно комбинировать множественные условия с помощью логических операций, и какие ошибки чаще всего допускают начинающие программисты при написании сложных логических выражений?

8. Перечислите не менее трёх типовых задач, решаемых с помощью ветвлений, и кратко опишите логику выбора ветви для каждой из них.

Практические занятия:

Задание 1. Напишите программу, которая получает от пользователя число и выводит на экран сообщение "Число четное", если число четное, и "Число нечетное", если число нечетное.

Задание 2. При регистрации на сайтах требуется вводить пароль дважды. Это сделано для безопасности, поскольку такой подход уменьшает возможность неверного ввода пароля.

Напишите программу, которая сравнивает пароль и его подтверждение. Если они совпадают, то программа выводит: «Пароль принят», иначе: «Пароль не принят».

Задание 3. На вход программе подаётся целое число — возраст пользователя. Программа должна вывести текст «Доступ разрешен», если возраст не менее 18, и «Доступ запрещен» - в противном случае.

Задание 4. Напишите программу, которая определяет наименьшее из двух чисел. На вход программе подаётся два различных целых числа. Программа должна вывести наименьшее из двух чисел.

Задание 5. Напишите программу, которая считывает три числа и подсчитывает сумму только положительных чисел.

Задание 6. Напишите программу, которая получает от пользователя его имя, фамилию и возраст, и выводит на экран сообщение: "Привет, [имя] [фамилия]! Вам [возраст] лет", если возраст меньше 18 лет, и "Здравствуйте, [имя] [фамилия]! Вам [возраст] лет", если возраст больше или равен 18 лет.

Задание 7. Напишите программу, которая получает от пользователя его возраст и выводит на экран сообщение: "Вам нужно идти в детский сад", если возраст меньше 6 лет, "Вам нужно идти в школу", если возраст меньше 18 лет, и "Вам нужно работать", если возраст больше 18 лет.

Задание 8. Даны три целых числа. Определите, сколько среди них совпадающих. Программа должна вывести одно из чисел: 3 (если все совпадают), 2 (если два совпадает) или 0 (если все числа различны).

Задание 9. Напишите программу, которая получает от пользователя два числа и выводит на экран сообщение "Оба числа положительные", если оба числа являются положительными, "Оба числа отрицательные", если оба числа являются отрицательными, и "Одно число положительное, а другое - отрицательное", если оба числа разных знаков.

Задание 10. Напишите программу, которая проверяет, что все три цифры натурального трёхзначного числа различны.

Задание 11. Напишите программу, которая по координатам точки, не лежащей на осях координат, определяет номер координатной четверти, в которой она находится.

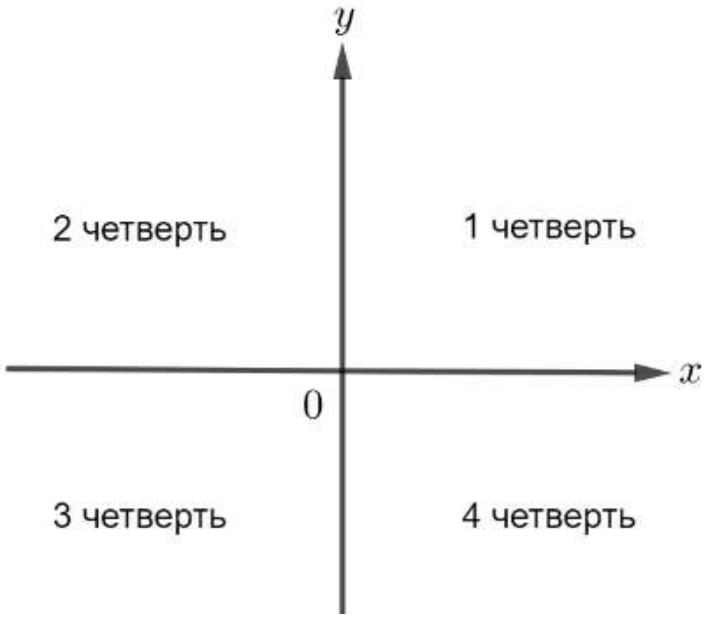

Рис. 3. Координатная ось

Задание 12. Напишите программу, которая принимает целое число x и определяет, принадлежит ли данное число указанному промежутку.

Рис. 4. Координатная ось

Задание 13. Напишите программу, которая принимает целое число x и определяет, принадлежит ли данное число указанным промежуткам.

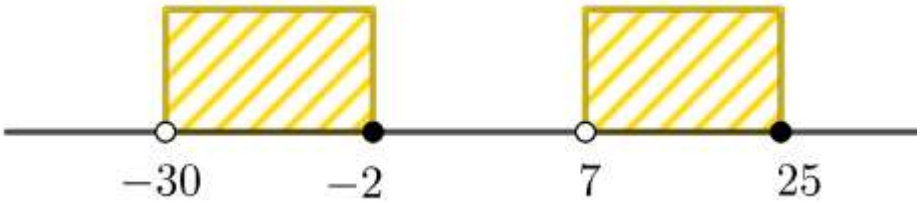

Рис. 5. Координатная ось

Задание 14. Напишите программу, которая принимает целое число x и определяет, принадлежит ли данное число указанным промежуткам.

Рис. 6. Координатная ось

Задание 15. Назовем число красивым, если оно является четырехзначным и делится нацело на 7 или на 17. Напишите программу, определяющую, является ли введённое число красивым. Программа должна вывести «YES», если число является красивым, или «NO» в противном случае.

Задание 16. Напишите программу, которая принимает три положительных числа и определяет, существует ли невырожденный треугольник с такими сторонами. Треугольник существует, если выполняется неравенство треугольника:

$$a + b > c$$
$$a + c > b$$
$$b + c > a,$$

Рис. 7. Неравенство существования треугольника

где a,b,c - стороны треугольника.

Задание 17. Напишите программу, которая определяет, является ли год с данным номером високосным. Если год является високосным, то выведите «YES», иначе выведите «NO». Год является високосным, если его номер кратен 4, но не кратен 100, или если он кратен 400.

Задание 18. Напишите программу, которая принимает три положительных числа и определяет вид треугольника, длины сторон которого равны введенным числам. Программа должна вывести на экран текст – вид треугольника («Равносторонний», «Равнобедренный» или «Разносторонний»).

Задание 19. Дан порядковый номер месяца (1,2,…, 12). Напишите программу, которая выводит на экран количество дней в этом месяце. Принять, что год является невисокосным.

Задание 20. Известен вес боксера-любителя (целое число). Известно, что вес таков, что боксер может быть отнесён к одной из трех весовых категорий: Легкий вес – до 60 кг (невключительно); Первый полусредний вес – до 64 кг (невключительно); Полусредний вес – до 69 кг (невключительно). Напишите программу, определяющую, в какой категории будет выступать данный боксер.

Задание 21. Напишите программу, которая считывает с клавиатуры два целых числа и строку. Если эта строка является обозначением одной из четырёх математических операций (+, -, *, /), то выведите результат применения этой операции к введённым ранее числам, в противном случае выведите «Неверная операция» (без кавычек). Если пользователь захочет поделить на ноль, выведите текст «На ноль делить нельзя!»

Задание 22. Напишите программу для вычисления и оценки индекса массы тела (ИМТ) человека. ИМТ показывает, человек весит больше или меньше нормы для своего роста. ИМТ человека рассчитывают по формуле:

ИМТ=масса (кг)рост(м)×рост(м), ИМТ=рост(м)×рост(м)масса (кг),где масса измеряется в килограммах, а рост – в метрах.

Масса человека считается оптимальной, если его ИМТ находится между 18.518.5 и 2525. Если ИМТ меньше 18.518.5, то считается, что человек весит ниже нормы. Если значение ИМТ больше 2525, то считается, что человек весит больше нормы. Программа должна вывести "Оптимальная масса", если ИМТ находится между 18.518.5 и 2525 (включительно). "Недостаточная масса", если ИМТ меньше 18.518.5 и "Избыточная масса", если значение ИМТ больше 2525.

Задание 23. Напишите программу, которая помогает игроку определить, к какому классу героя из фэнтезийной RPG он бы относился. Пользователю предлагается выбрать сочетание нескольких характеристик (например, сила, ловкость, мудрость). В зависимости от выбранного сочетания, программа выводит сообщение о принадлежности к определенному классу героя, например: "Ты мощный воин!" или "Ты мудрый маг!", или "Ты ловкий вор"

Тема 4. Циклические алгоритмы, операторы циклов Python

Контрольные вопросы:

1. Дайте определение цикла в программировании, назовите три обязательных компонента любого цикла и поясните роль каждого.
2. В чём заключается принципиальная разница между циклом «while» и циклом «for», и в каких ситуациях предпочтительно использовать каждый из них?
3. Как работает итерация по диапазону чисел в цикле «for», что такое функция генерации диапазона, и как с её помощью организовать перебор чисел с разным шагом?
4. Для чего служат операторы управления циклом «break» и «continue», и чем отличается их действие на выполнение цикла?
5. Что такое бесконечный цикл, каковы типичные причины его возникновения, и какие существуют способы предотвратить или корректно прервать бесконечный цикл?
6. Объясните, что такое вложенные циклы, приведите пример задачи (например, вывод таблицы умножения или числового паттерна), где вложенные циклы необходимы.
7. Как организовать перебор символов строки с помощью цикла «для», и чем этот способ отличается от использования индексов и операции среза?
8. Перечислите не менее трёх типовых задач, решаемых с помощью циклов, и поясните, как меняется состояние переменных от итерации к итерации в каждой из них.

Практические занятия:

Задание 1. На вход программе подается натуральное число n. Напишите программу, которая печатает звездный прямоугольник размерами n×19.

Задание 2. Даны два целых числа m и n. Напишите программу, которая выводит все целые числа от m до n включительно в порядке возрастания, если m<n, или в порядке убывания в противном случае.

Задание 3. Даны два натуральных числа m≤n. Напишите программу, которая выводит все целые числа от m до n включительно, удовлетворяющие хотя бы одному из условий:

– число кратно 17;

– число оканчивается на 9;

– число кратно 3 и 5 одновременно.

Задание 4. Напишите программу, которая считывает 10 чисел и выводит произведение отличных от нуля чисел.

Задание 5. На вход программе подается натуральное число n. Напишите программу, которая вычисляет сумму всех его делителей.

Задание 6. Напишите программу, которая считывает последовательность из 10 целых чисел и определяет является ли каждое из них четным или нет. Программа должна вывести строку «YES», если все числа четные и «NO» в ином случае.

Задание 7. На вход программе подается последовательность слов, каждое слово на отдельной строке. Концом последовательности является одно из трех слов: «стоп», «хватит», «достаточно» (маленькими буквами, без кавычек). Сами эти слова в последовательность не входят, лишь символизируя её окончание. Напишите программу, которая выводит общее количество членов данной последовательности.

Задание 8. Ограничить ввод данных в строку, чтобы при просьбе ввести паспорт, программа просила снова и снова исправить число, пока то не будет равно девяти символам

Задание 9. На вход программе подается последовательность целых чисел, каждое число на отдельной строке. Признаком окончания последовательности является любое отрицательное число, при этом в саму последовательность оно не входит. Напишите программу, которая выводит сумму всех членов данной последовательности.

Задание 10. На вход программе подается последовательность целых чисел от 1 до 5, характеризующее оценку ученика, каждое число на отдельной строке. Концом последовательности является любое отрицательное число либо число, большее 5. Напишите программу, которая выводит количество пятерок.

Задание 11. Дано натуральное число. Напишите программу, которая выводит его цифры в столбик в обратном порядке.

Задание 12. Дано натуральное число n ≥ 10. Напишите программу, которая определяет его максимальную и минимальную цифры.

Задание 13. Дано натуральное число. Напишите программу, которая вычисляет:

– сумму его цифр;

– количество цифр в нем;

– произведение его цифр;

– среднее арифметическое его цифр;

– его первую цифру;

– сумму его первой и последней цифры.

Задание 14. Дано натуральное число n. Напишите программу, которая печатает численный треугольник с высотой равной n, в соответствии с примером:

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21

РАЗДЕЛ 2. ИСПОЛЬЗОВАНИЕ СТРОК И СПИСКОВ

Тема 5. Строки в Python. Функции и методы работы со строками

Контрольные вопросы:

1. Почему строки в Python называют «неизменяемой последовательностью символов», и какие ограничения накладывает это свойство на операции со строками?
2. Что такое прямая и обратная индексация в строках, чем отрицательные индексы удобны, и какой индекс соответствует последнему символу строки?
3. Объясните, что такое операция среза, из каких трёх параметров она состоит, и что произойдёт, если опустить каждый из них?
4. Перечислите не менее четырёх методов преобразования регистра строки и поясните, чем метод отличается от простой функции.
5. Какие методы позволяют удалить лишние пробелы по краям строки, разбить строку на части по заданному разделителю и, наоборот, склеить список строк в одну строку с разделителем?
6. Как выполняется поиск подстроки в строке, какой метод для этого используется, и что он возвращает в случае успеха и в случае неудачи?
7. Какие методы существуют для проверки свойств строки, и в каких задачах такие проверки критически важны?
8. Приведите пример типовой задачи на обработку строк и опишите последовательность шагов для её решения.

Практические занятия:

Задание 1. На вход программе подается одна строка. Напишите программу, которая выводит в столбик элементы строки в обратном порядке. Срезы не использовать.

Задание 2. На вход программе подается одна строка. Напишите программу, которая определяет, сколько раз в строке встречаются символы + и *. Метод count не использовать

Задание 3. На вход программе подается одна строка с буквами русского языка. Напишите программу, которая определяет количество гласных и согласных букв. Используйте для проверки строки:

- 'ауоыиэяюёеАУОЫИЭЯЮЁЕ';
- 'бвгджзйклмнпрстфхцчшщБВГДЖЗЙКЛМНПРСТФХЦЧШЩ'.

Задание 4. Вывести первые 10 символов строки s='Календарь Майя'.

Задание 5. Вывести последние 5 символов строки s='Календарь Майя'.

Задание 6. Используя срезы, вывести каждый 5 символ строки s (начиная с 0-го индекса).

s=' Жизнь предназначена не только для учебы, но если вы не можете справиться с этой частью жизни, то с чем тогда вы способны справиться?'

Задание 7. Проверить, является ли введённая строка полиндромом (читается одинаково в обе стороны - шалаш, казак, потоп).

Задание 8. На вход программе подается одна строка. Напишите программу, которая выводит:

- общее количество символов в строке;
- исходную строку, повторенную 3 раза;
- первый символ строки;
- первые три символа строки;
- последние три символа строки;
- строку в обратном порядке;
- строку с удаленным первым и последним символом.

Задание 9. Даны два четырёхзначных числа A и B. Выведите все четырёхзначные числа на отрезке от A до B, запись которых является палиндромом.

Задание 10. На вход программе подается одна строка. Напишите программу, которая выводит:

- третий символ этой строки;
- предпоследний символ этой строки;
- первые пять символов этой строки;
- всю строку, кроме последних двух символов;
- все символы с четными индексами;
- все символы с нечетными индексами;
- все символы в обратном порядке;
- все символы строки через один в обратном порядке, начиная с последнего.

Задание 11. В строке s='Проффессор' исправьте ошибку

Задание 12. Удалить из строки первый и последний символ с помощью среза

Задание 13. Вводится строка текста. Напишите программу, которая подсчитывает количество цифр в данной строке.

Задание 14. Дана строка текста. Напишите программу для подсчета стоимости строки, исходя из того, что один любой символ (в том числе пробел) стоит 60 копеек. Ответ дайте в рублях и копейках.

Задание 15. Введенное существительное в единственном числе преобразуйте в множественное, используя правила:

- если слово оканчивается на 'es', то заменяем последние две буквы на 'e' (apples - apple, boxes - boxe)
- слова, которые заканчиваются на согласный звук и 'y', 'y' заменяем на 'ies' (lorry – lorries; family – families; fly – flies)
- к словам заканчивающиемся на ch, x, sh, ss, s добавляем окончание «-es» (match – matches; box – boxes; brush – brushes; glass – glasses; bus – buses)
- для слов заканчивающихся на «f», «fe». Вместо «f» и «fe» → «ves» (wolf – wolves; wife – wives)
- во всех остальных случаях просто добаввляем в конце s (boy – boys; guy – guys; pear – pears; student – students; cup – cups)

Задание 16. На платформе пользователи оставляют комментарии, но не все из них соответствуют правилам. Так, например, модератор Сэм считает неуместными комментариями те, которые представляют собой пустую строку или состоят только из пробелов. Подобные комментарии он удаляет.

Ваша задача – написать программу, которая поможет Сэму проверять комментарии. Программа должна принимать на вход натуральное число n, а затем n строк, представляющих тексты комментариев. Для каждого комментария ваша программа должна выводить номер этого комментария (начиная с 1), затем двоеточие (:), затем через пробел его текст или сообщение «COMMENT SHOULD BE DELETED» (без кавычек), если комментарий должен быть удалён Сэмом.

Задание 17. В службе по дорожному движению решили оптимизировать процесс создания автомобильных номеров: вместо человека генерацию автомобильных номеров поручили некоторой GPT (модели машинного обучения). Как мы знаем, искусственный интеллект ещё сыроват и делает много ошибок, поэтому его результаты следует тщательно проверять. Напишите программу, которая принимает на вход строку и проверяет, является ли эта строка корректным автомобильным номером. Программа должна вывести «YES» (без кавычек), если искусственный интеллект справился со своей задачей, или «NO» (без кавычек) в противном случае. В нашей задаче корректным автомобильным номером будем считать следующие форматы:

<БУКВА><ЦИФРА><ЦИФРА><ЦИФРА><БУКВА><БУКВА><ЦИФРА><ЦИФРА>

<БУКВА><ЦИФРА><ЦИФРА><ЦИФРА><БУКВА><БУКВА><ЦИФРА><ЦИФРА><ЦИФРА>

где <ЦИФРА> – это любая цифра, а <БУКВА> – это одна из букв кириллицы АВЕКМНОРСТУХ.

Задание 18. Необходимо решить задачу на валидацию имени пользователя. Пользователь пытается создать никнейм для своего аккаунта в соцсети Y. Правила для корректного никнейма в соцсети Y следующие:

- никнейм должен начинаться с символа @

– никнейм должен содержать от 5 до 15(включительно) символов (включая первый символ @)

– никнейм должен содержать только строчные буквы и цифры (помимо первого символа @)

Напишите программу, которая выводит «Correct» (без кавычек), если никнейм соответствует всем вышеприведенным правилам, или «Incorrect» (без кавычек) в противном случае.

Примечание. Обратите внимание, что никнейму необязательно содержать строчные буквы и цифры одновременно, никнейм может содержать только строчные буквы или только цифры (помимо первого символа @). Например, следующие никнеймы считаются корректными:

@duncan

@1111

Задание 19. На вход программе подаются два целых числа a и b. Ваша программа должна посчитать для этих чисел сумму их кубов и куб их суммы и вывести результат вычислений в следующем формате:

Для чисел <число a> и <число b>:

Сумма кубов: <число a>**3 + <число b>**3 = <сумма кубов a и b>

Куб суммы: (<число a> + <число b>)**3 = <куб суммы a и b>

Тема 6. Списки. Списочные выражения

Контрольные вопросы:

1. Чем список в Python отличается от строки с точки зрения изменяемости, и какие возможности появляются у программиста благодаря этому отличию?
2. Перечислите не менее трёх способов создания списков, и кратко охарактеризуйте каждый.
3. Что такое генератор списков, как он записывается, и в каких случаях его использование делает код более компактным и читаемым по сравнению с циклом?
4. Какие операции над списками поддерживаются в Python, и как результат конкатенации отличается от результата метода расширения?
5. Назовите не менее пяти методов списка и поясните действие каждого.
6. Что делают встроенные функции получения длины, суммы, минимального и максимального элемента применительно к спискам, и для каких типов данных эти функции работают корректно?
7. Объясните, что такое вложенные списки, как получить доступ к элементу на пересечении определённой строки и столбца, и как перебрать все элементы матрицы с помощью вложенных циклов?
8. В чём заключается проблема поверхностного копирования списков, чем глубокое копирование отличается от поверхностного, и в каких ситуациях необходимо применять именно глубокое копирование?

Практические занятия:

Задание 1. На вход программе подается одно число n. Напишите программу, которая выводит список [1, 2, 3, ..., n] и его сумму

Задание 2. Написать программу, которая запрашивает у пользователя четное число,n≥2 и выводит список четных чисел [2, 4, 6, ..., n] и его длину

Задание 3. Приведён следующий код:

primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71]

Дополните его, чтобы он:

- Вывел длину списка;
- Вывел сумму, минимум и максимум;
- Вывел последний элемент списка;
- Вывел список в обратном порядке (вспоминаем срезы);
- Вывел YES, если список содержит числа 5 и 17, и NO в противном случае;
- Вывел список с удаленными первым и последним элементами.

Задание 4. В программе вводится натуральное число n, а затем n строк. Напишите программу, которая создает из указанных строк список и выводит его в обратном порядке.

Задание 5. Дополните приведённый ниже код, используя списочное выражение, так, чтобы получить новый список, содержащий строки исходного списка, где у каждой строки удалён первый символ.

keywords = ['False', 'True', 'None', 'and', 'with', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'try', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'while', 'yield']

new_keywords =

print(new_keywords)

Задание 6. Дополните приведённый ниже код, используя списочное выражение, так, чтобы получить новый список, содержащий длины строк исходного списка.

keywords = ['False', 'True', 'None', 'and', 'with', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'try', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'while', 'yield']

lengths =

print(lengths)

Задание 7. Дополните приведённый ниже код, используя списочное выражение, так, чтобы получить новый список, содержащий только слова длиной не менее пяти символов (включительно).

keywords = ['False', 'True', 'None', 'and', 'with', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'try', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'while', 'yield']

new_keywords =

print(new_keywords)

Задание 8. Дополните приведённый ниже код, используя списочное выражение, так, чтобы получить список целых чисел-палиндромов от 100 до 1000.

```
palindromes = []
print(palindromes)
```

Задание 9. На вход программе подаётся натуральное число n. Напишите программу, использующую списочное выражение, которая создаёт список, содержащий квадраты чисел от 1 до n (включительно), а затем выводит его элементы построчно, то есть каждый на отдельной строке.

Задание 10. На вход программе подаётся строка текста, содержащая целые числа. Напишите программу, использующую списочное выражение, которая выведет кубы указанных чисел на одной строке.

Задание 11. На вход программе подаётся строка текста, содержащая слова. Напишите программу, которая выводит слова введённой строки в столбик.

Задание 12. На вход программе подаётся строка текста. Напишите программу, использующую списочное выражение, которая выводит все цифровые символы данной строки.

Задание 13. На вход программе подаётся строка текста, содержащая целые числа. Напишите программу, использующую списочное выражение, которая выведет квадраты чётных чисел, кроме тех квадратов, которые оканчиваются на цифру 4.

Тема 7. Генерация случайных чисел в программировании задач моделирования и игр.

Контрольные вопросы:

1. Почему числа, генерируемые компьютером, называют не случайными, а псевдослучайными, и в чём заключается принципиальная разница между истинной случайностью и псевдослучайностью?
2. Какая стандартная библиотека в Python отвечает за генерацию случайных чисел, и какие функции этого модуля являются наиболее востребованными на практике?
3. Чем отличаются функции получения случайного целого числа в заданном диапазоне и получения случайного целого числа с указанным шагом, и в каких задачах удобнее использовать каждую из них?
4. Как получить случайное вещественное число в диапазоне от нуля до единицы, и как, используя эту функцию, получить случайное вещественное число в любом другом произвольном диапазоне?
5. Какие функции модуля случайных чисел позволяют случайным образом выбрать один элемент из последовательности или случайным образом перемешать все элементы последовательности?
6. Что такое функция получения случайной выборки заданного размера, и чем она отличается от однократного случайного выбора элемента?

7. Для чего нужна инициализация генератора случайных чисел с помощью начального значения, и в каких ситуациях это бывает критически важно?

8. Приведите пример игровой или модельной задачи, и опишите, какие именно функции генерации случайных чисел и в какой последовательности применяются для её решения.

Практические занятия:

Задание 1. Напишите программу, которая моделирует броски монеты. Программа принимает на вход количество попыток и выводит результаты бросков: Орел или Решка (каждое на отдельной строке). Примечание. Например, при n=7 ваша программа может выводить:

Орел
Решка
Решка
Орел
Орел
Орел
Решка

Задание 2. Напишите программу, которая с помощью модуля random моделирует броски игрального кубика с 6 гранями. Программа принимает на вход количество попыток и выводит результаты бросков — выпавшее число, которое написано на грани кубика (каждое на отдельной строке). Примечание. Например, при n=7 ваша программа может выводить:

5
5
6
6
2
6
2

Задание 3. Напишите программу, которая с помощью модуля random генерирует случайный пароль. Программа принимает на вход длину пароля и выводит случайный пароль, содержащий только символы английского алфавита a..z, A..Z (в нижнем и верхнем регистре).

Примечание 1. Символам A..Z английского языка соответствуют номера с 65 по 90 в таблице символов ASCII.

Примечание 2. Символам a..z английского языка соответствуют номера с 97 по 122 в таблице символов ASCII.

Примечание 3. Используйте функцию chr() для получения символа по его номеру в таблице символов ASCII.

Примечание 4. Например, при длине пароля, равной 15 символам, ваша программа может выводить:

peJFAmhqfaAeKDu

Задание 4. Лотерейный билет содержит 7 чисел из диапазона от 1 до 49 (включительно). Напишите программу, которая с помощью модуля random генерирует 7 различных случайных чисел для лотерейного билета. Программа должна вывести числа в порядке возрастания на одной строке через один символ пробела. Убедитесь, что сгенерированные числа не содержат дубликатов.

Задание 5. IP адрес состоит из четырех чисел из диапазона от 0 до 255 (включительно) разделенных точкой. Напишите функцию generate_ip(), которая с помощью модуля random генерирует и возвращает случайный корректный IP адрес. Пример правильного (неправильного) IP адреса:

```
192.168.5.250      # правильный
199.300.521.255    # неправильный
```

Задание 6. Почтовый индекс в Латверии имеет вид: LetterLetterNumber_NumberLetterLetter, где Letter – заглавная буква английского алфавита, Number – число от 0 до 99 (включительно). Напишите функцию generate_index(), которая с помощью модуля random генерирует и возвращает случайный корректный почтовый индекс Латверии. Пример правильного (неправильного) индекса Латверии:

```
AB23_56VG      # правильный
V3F_231GT      # неправильный
```

Задание 7. Напишите программу, которая с помощью модуля random перемешивает случайным образом содержимое матрицы (двумерного списка).

```
matrix = [[1, 2, 3, 4],
          [5, 6, 7, 8],
          [9, 10, 11, 12],
          [13, 14, 15, 16]]
```

Задание 8. Напишите программу, которая с помощью модуля random генерирует 100 случайных номеров лотерейных билетов и выводит их каждый на отдельной строке. Обратите внимание, вы должны придерживаться следующих условий:

- номер не может начинаться с нулей;
- номер лотерейного билета должен состоять из 7 цифр;
- все 100 лотерейных билетов должны быть различными.

Задание 9. Анаграмма – это слово образованное путём перестановки букв, составляющих другое слово. Например, слова пила и липа или пост и стоп – анаграммы. Напишите программу, которая считывает одно слово и выводит с помощью модуля random его случайную анаграмму. Обратите внимание на то, что метод shuffle() работает со списком, а не со строкой.

Задание 10. Для игры в бинго требуется карточка размером 5×5, содержащая различные (уникальные) целые числа от 1 до 75 (включительно), при этом центральная клетка является пустой (она заполняется числом 0). Напишите программу, которая с помощью

модуля random генерирует и выводит случайную карточку для игры в бинго. Пример возможного ответа:

```
1  16 31 46 61
10 30 42 47 68
3  18 0  48 63
9  19 34 49 70
5  20 35 50 65
```

Примечание. Для наглядности рекомендуем отводить на вывод каждого числа ровно 3 символа. Для этого используйте строковый метод ljust().

Задание 11. Напишите программу, которая с помощью модуля random генерирует n паролей длиной m символов, состоящих из строчных и прописных английских букв и цифр, кроме тех, которые легко перепутать между собой:

- «l» (L маленькое);
- «I» (i большое);
- «1» (цифра);
- «o» и «O» (маленькая и большая буквы);
- «0» (цифра). На вход программе подаются два числа n и m, каждое на отдельной строке. Программа должна вывести n паролей длиной m символов в соответствии с условием задачи, каждый на отдельной строке.

Примечание 1. Считать, что числа n и m всегда таковы, что требуемые пароли сгенерировать возможно.

Примечание 2. В каждом пароле **необязательно** должна присутствовать хотя бы одна цифра и буква в верхнем и нижнем регистре.

Примечание 3. Решение задачи удобно оформить в виде двух вспомогательных функций:

- функция generate_password(length) – возвращает случайный пароль длиной length символов;
- функция generate_passwords(count, length) – возвращает список, состоящий из count случайных паролей длиной length символов.

Пример:

```
def generate_password(length):
    pass
def generate_passwords(count, length):
    pass
n, m = int(input()), int(input())
```

Задание 12. Напишите программу, которая случайным образом назначает каждому ученику его тайного друга, который будет вместе с ним решать задачи по программированию.

- На вход программе в первой строке подается число n – общее количество учеников. Далее идут n строк, содержащих имена и фамилии учеников.

– Программа должна вывести имя и фамилию ученика (в соответствии с исходным порядком) и имя и фамилию его тайного друга, разделённые дефисом.

– Обратите внимание, что нельзя быть тайным другом самому себе и нельзя быть тайным другом для нескольких учеников.

Образцы возможного ввода и вывода:

Sample Input:

3
Светлана Зуева
Аркадий Белых
Борис Боков

Sample Output:

Светлана Зуева - Борис Боков
Аркадий Белых - Светлана Зуева
Борис Боков - Аркадий Белых

5.3. Один или несколько тематических блоков дисциплины завершаются контрольной работой по разделу (далее – КР). Текущий контроль успеваемости по дисциплине предусматривает не менее 2 (двух) и не более 10 (десяти) КР в течение периода освоения дисциплины.

Максимальное количество баллов за любой тип работ в рамках КР составляет 100 (сто) баллов.

Распределение весовых коэффициентов по КР в рамках текущего контроля успеваемости по дисциплине и формулы расчета:

Наименование контрольной работы	Максимальное количество баллов за работу в рамках КР, которое может набрать студент	Коэффициент веса контрольной работы	Результат контрольной работы, участвующий в формировании итоговой балльной оценки по дисциплине
КР 1	100	0,15	15
КР 2	100	0,15	15
Итого:	x	0,30	30

Формула расчета результата контрольной работы:

Результат контрольной работы = Количество баллов за работу в рамках КР X Коэффициент веса контрольной работы.

5.4. Формы текущего контроля успеваемости обучающихся в рамках КР и типовые оценочные материалы:

КР-1

Раздел 1. Введение в алгоритмизацию на Python

Задание 1. Перечислите пять основных свойств алгоритма, раскройте суть каждого из них. Объясните, почему язык Python относится к интерпретируемым языкам с динамической типизацией, и приведите одно преимущество и один недостаток такой организации по сравнению с компилируемыми языками со статической типизацией.

Задание 2. Напишите программу, которая запрашивает у пользователя его имя (строку) и год рождения (целое число). Программа должна вывести на экран приветствие вида: «Привет, [имя]! В этом году тебе исполняется [текущий_год − год_рождения] лет». Для получения текущего года используйте встроенные средства. Все преобразования типов должны быть выполнены явно.

Задание 3. Даны три переменные, содержащие целые числа. Не используя дополнительные переменные, поменяйте местами значения первой и третьей переменных. Запишите последовательность операций присваивания. После выполнения выведите на экран новые значения всех трёх переменных.

Задание 4. Напишите программу, которая запрашивает у пользователя три числа. Программа должна определить и вывести на экран:

- являются ли все три числа положительными;
- является ли хотя бы одно из чисел отрицательным;
- равно ли хотя бы одно из чисел нулю.

Для каждого из трёх условий программа должна выдать отдельное сообщение. Используйте вложенные или составные условные конструкции с логическими операциями.

Задание 5. Составьте блок-схему алгоритма решения квадратного уравнения вида $a \cdot x^2 + b \cdot x + c = 0$. На вход подаются коэффициенты a, b, c (вещественные числа). Алгоритм должен вычислить дискриминант и в зависимости от его значения вывести:

- два различных корня;
- один корень;
- сообщение о том, что действительных корней нет.

Отдельно предусмотрите случай, когда $a = 0$. Блок-схему нарисуйте от руки или с помощью графических средств, результат сфотографируйте или сохраните в виде изображения.

Задание 6. Напишите программу, которая с помощью цикла «while» вычисляет сумму всех натуральных чисел от 1 до N, где N вводится пользователем. Программа также должна подсчитать количество выполненных итераций и вывести его на экран вместе с суммой. Предусмотрите случай, если пользователь ввёл число меньше 1 — в таком случае программа должна сообщить об ошибке и не выполнять вычислений.

Задание 7. Используя цикл «for» и функцию генерации диапазона, выведите на экран следующий числовой паттерн для произвольного нечётного числа N. Пример для N = 5:

1
22
333
4444
55555

Программа должна корректно работать для любых нечётных N от 1 до 9. Объясните, почему для чётных N такой паттерн будет выглядеть иначе.

Задание 8. Напишите программу «Угадай число», работающую по следующим правилам:

- константой задаётся целое число в диапазоне от 1 до 100;
- пользователь вводит свои варианты чисел с клавиатуры;
- после каждой попытки программа сообщает: «больше», «меньше» или «угадал»;
- программа ведёт счётчик попыток;
- когда число угадано, программа поздравляет пользователя и показывает количество затраченных попыток;
- если пользователь ввёл число вне диапазона от 1 до 100, программа выдаёт предупреждение, но попытка всё равно засчитывается.

КР-2

Раздел 2. Использование строк и списков

Задание 1. Напишите программу, которая запрашивает у пользователя строку, состоящую из нескольких слов, разделённых пробелами. Программа должна выполнить следующие действия:

- удалить лишние пробелы в начале и в конце строки;
- разбить строку на список отдельных слов;
- определить количество слов в строке;
- заменить все вхождения буквы «а» (в любом регистре) на символ «@»;
- вывести полученную преобразованную строку на экран.

Задание 2. Дана строка, содержащая текст на русском языке. Пользователь вводит с клавиатуры два слова (искомое и замену). Программа должна найти в исходной строке все вхождения искомого слова (как отдельного слова, а не части другого слова) и заменить их на слово-замену. При поиске учитывайте, что слова могут быть разделены пробелами, знаками препинания или находиться в начале/конце строки. Регистр букв учитывать не следует. Для сравнения строк без учёта регистра используйте соответствующие методы преобразования регистра.

Задание 3. Создайте список из 10 целых чисел, введённых пользователем с клавиатуры. Программа должна выполнить следующие операции:

- вывести исходный список;
- вывести третий элемент списка и предпоследний элемент;

– вывести срез, содержащий элементы со второго по шестой включительно;

– добавить в конец списка число, равное сумме всех элементов исходного списка;

– удалить из списка все элементы, значение которых меньше 5;

– отсортировать оставшиеся элементы по возрастанию и вывести итоговый список.

Задание 4. С помощью генератора списков создайте и выведите на экран следующие списки:

– список квадратов всех целых чисел от 1 до 20;

– список, содержащий только чётные числа из диапазона от 10 до 50;

– список, в котором каждый элемент исходного списка строк преобразован в его длину;

– список, состоящий из 15 случайных целых чисел в диапазоне от 0 до 100.

Для последнего пункта поясните в комментарии, чем генератор списков удобнее классического цикла с методом добавления в конец.

Задание 5. Напишите программу, которая создает матрицу размером 5×5, заполненную случайными целыми числами в диапазоне от 1 до 50. Программа должна:

– вывести исходную матрицу в виде таблицы;

– вычислить и вывести сумму всех элементов матрицы;

– вычислить и вывести среднее арифметическое всех элементов матрицы;

– найти и вывести максимальный элемент и его координаты (номер строки и столбца, считая с единицы);

– заменить все элементы главной диагонали на ноль и вывести преобразованную матрицу.

Задание 6. Создайте программу – симулятор подбрасывания двух игральных кубиков. Каждый кубик может показать целое число от 1 до 6. Программа должна выполнить N подбрасываний. Для каждого подбрасывания программа генерирует два случайных числа, вычисляет их сумму и записывает эту сумму в отдельный список. После завершения всех подбрасываний программа выводит:

– список всех полученных сумм;

– сколько раз выпала каждая возможная сумма в виде гистограммы (например: «Сумма 2: *****» – пять звёздочек означает пять выпадений);

– наиболее часто выпадавшую сумму.

Используйте функции модуля случайных чисел для генерации результатов бросков.

Задание 7. Напишите программу «Генератор безопасных паролей». Программа запрашивает у пользователя желаемую длину пароля. После этого программа генерирует случайный пароль, который обязательно должен содержать:

– хотя бы одну заглавную латинскую букву (от A до Z);

- хотя бы одну строчную латинскую букву (от a до z);
- хотя бы одну цифру (от 0 до 9);
- хотя бы один специальный символ из набора: ! @ # $ % ^ & * ( ).

Программа должна гарантировать выполнение всех условий. Полученный пароль выводится на экран. Для работы со строками и списками используйте методы конкатенации, преобразования регистра и проверки принадлежности символа к набору.

Задание 8. Реализуйте программу «Моделирование подбрасывания монеты методом Монте-Карло». Программа запрашивает у пользователя количество экспериментов (K – целое число, не менее 1000). В каждом эксперименте моделируется подбрасывание правильной монеты 10 раз (орёл или решка с равной вероятностью). Программа подсчитывает, в скольких экспериментах орёл выпал ровно 5 раз. После завершения всех экспериментов программа вычисляет и выводит:

- абсолютное количество экспериментов, в которых орёл выпал ровно 5 раз;
- вероятность такого события в процентах.

6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине

6.1. Промежуточная аттестация по дисциплине «Информатика и программирование» проводится в форме зачета с оценкой в первом семестре в письменной форме. Обучающийся получает три теоретических вопроса и одно практическое задание.

Теоретические вопросы направлены на проверку:

- понимания базовых понятий алгоритмизации;
- знания синтаксических конструкций и семантики языка Python;
- понимания принципов генерации псевдослучайных чисел и их применения в задачах моделирования и разработке игр;
- знания методов обработки строк;
- понимания структуры и возможностей списков как изменяемой коллекции, включая списочные выражения и вложенные списки;
- умения логически рассуждать о работе фрагментов кода и предсказывать результат их выполнения без компьютера;
- понимания различий между циклами, способов управления ими.

Практическое задание направлено на проверку умений:

- разрабатывать алгоритмы решения типовых вычислительных и логических задач;
- реализовывать программы на Python с использованием ветвлений и циклов различных типов;
- применять методы работы со строками для обработки текстовой информации;
- создавать и модифицировать списки, использовать списочные выражения для компактной записи преобразований данных;
- генерировать случайные числа и использовать их в игровых и модельных задачах;
- читать и анализировать готовый код, находить и исправлять ошибки;
- оформлять программный код с комментариями и понятными именами переменных в соответствии с принципами читаемости;
- тестировать программы на различных наборах входных данных, включая граничные случаи.

6.2. Типовые оценочные материалы промежуточной аттестации.

Вопросы к зачету с оценкой:

1. Дайте определение понятию «алгоритм». Перечислите и раскройте пять основных свойств алгоритмов.
2. Опишите все известные вам способы описания алгоритмов, укажите их достоинства и недостатки.

3. В чём заключается разница между интерпретируемыми и компилируемыми языками программирования? К какой группе относится Python?

4. Что такое динамическая типизация? Назовите её преимущества и недостатки по сравнению со статической типизацией.

5. Перечислите базовые типы данных в Python. Для каждого типа приведите пример литерала (способа записи в коде).

6. Каким образом организован ввод данных с клавиатуры и вывод на экран в Python? Опишите способы форматирования выводимых строк.

7. Что такое явное и неявное преобразование типов? В каких ситуациях Python выполняет преобразование автоматически, а когда требуется вмешательство программиста?

8. Перечислите все арифметические операции в Python. Чем отличается обычное деление от целочисленного? Что такое остаток от деления?

9. Какие операции сравнения существуют в Python? Какое значение (какого типа) возвращает каждая из них?

10. Опишите логические операции «и», «или», «не». Каков приоритет их выполнения по отношению друг к другу и к операциям сравнения?

11. Какие правила именования переменных существуют в Python? Какие имена являются допустимыми, а какие запрещёнными?

12. Опишите структуру условного оператора «если» (if). Для чего нужны блоки «иначе» (else) и «иначе если» (elif)?

13. Что такое вложенные условные конструкции? Приведите пример задачи, где они необходимы.

14. Что такое тернарный условный оператор? В каких случаях его применение оправдано?

15. Как в Python проверить принадлежность числа заданному диапазону? Приведите пример записи такого условия.

16. Перечислите типовые ошибки, которые возникают у начинающих программистов при работе с ветвлениями, циклами, строками и списками. Как их диагностировать и исправлять?

17. Дайте определение цикла. Какие три обязательных компонента содержит любой циклический алгоритм?

18. Опишите цикл «пока» (while): синтаксис, особенности применения, ситуации, когда он предпочтительнее цикла «для».

19. Опишите цикл «для» (for) в Python: синтаксис, работа с функцией генерации диапазона, перебор элементов последовательности.

20. Для чего служат операторы управления циклом «прервать» (break) и «продолжить» (continue)? Приведите примеры использования каждого.

21. Что такое бесконечный цикл? Каковы причины его возникновения и способы предотвращения?

22. Что такое вложенные циклы? Приведите пример задачи, решаемой с помощью вложенных циклов.

23. В чём отличие цикла с известным числом итераций от цикла с неизвестным? Приведите примеры задач для каждого типа.

24. Что такое строка в Python? Почему строки называют неизменяемыми последовательностями символов?

25. Объясните механизм индексации строк (прямая и обратная). Что такое срез строки, из каких параметров он состоит?

26. Перечислите не менее пяти методов работы со строками (преобразование регистра, удаление пробелов, разбиение, склеивание, поиск, замена). Опишите действие каждого.

27. Какие методы позволяют проверить свойства строки (состоит ли только из цифр, только из букв, является ли пробельной)?

28. Что такое список в Python? Чем список отличается от строки с точки зрения изменяемости?

29. Перечислите способы создания списков. Что такое генератор списков (списочное выражение)? Приведите пример.

30. Какие операции над списками поддерживаются в Python (объединение, повторение, индексация, срезы)?

31. Перечислите не менее пяти методов списка (добавление, вставка, удаление по значению, удаление по индексу, сортировка, разворот). Опишите действие каждого.

32. Какие встроенные функции (получение длины, суммы, минимального, максимального элемента) применимы к спискам?

33. Что такое вложенные списки (матрицы)? Как получить доступ к элементу по индексам строки и столбца?

34. В чём заключается проблема поверхностного копирования списков? Чем глубокое копирование отличается от поверхностного?

35. Сравните эффективность (по времени выполнения и читаемости кода) различных способов обработки списка: классический цикл с методом добавления в конец и генератор списков.

36. Почему числа, генерируемые компьютером, называют псевдослучайными? В чём отличие от истинной случайности?

37. Какой модуль в Python отвечает за генерацию случайных чисел? Назовите не менее четырёх его основных функций и опишите их назначение.

38. В чём разница между функциями получения случайного целого числа в диапазоне и случайного вещественного числа в диапазоне?

39. Что такое функция получения случайной выборки заданного размера? Чем она отличается от однократного случайного выбора элемента?

40. Для чего нужна инициализация генератора случайных чисел с помощью начального значения (функция seed)? Приведите примеры использования.

Пример практического задания

Необходимо разработать программу на языке Python, которая выполняет две независимые функции (пользователь выбирает режим работы в начале программы).

Режим 1. Анализатор текста

Программа запрашивает у пользователя текст (одна строка или несколько строк до ввода пустой строки). После ввода текста программа должна выполнить следующие действия:

1. Вывести базовую статистику:
   - подсчитать общее количество символов в тексте (включая пробелы);
   - подсчитать количество символов без учёта пробелов;
   - подсчитать количество слов (слова разделяются пробелами и знаками препинания – точкой, запятой, восклицательным и вопросительным знаками);
   - определить длину самого длинного слова (без учёта знаков препинания) и вывести это слово.
2. Проанализировать буквы:
   - перевести весь текст в нижний регистр;
   - удалить из текста все символы, не являющиеся буквами и пробелами;
   - определить топ-5 самых часто встречающихся букв в тексте;
   - вывести все буквы, которые не встречаются в тексте ни разу.
3. Провести поиск палиндромов:
   - найти в тексте все слова, которые читаются одинаково слева направо и справа налево (без учёта регистра, игнорируя знаки препинания). Примеры: «топот», «казак», «доход»;
   - вывести найденные палиндромы (каждый с новой строки). Если палиндромов нет — вывести соответствующее сообщение.

Режим 2. Генератор паролей

Программа запрашивает у пользователя желаемую длину пароля (целое число от 8 до 24 включительно) и количество паролей для генерации (целое число от 1 до 10). Для каждого пароля программа должна:

1. Сгенерировать случайный пароль указанной длины, используя следующие символы:
   - строчные латинские буквы (a–z);
   - заглавные латинские буквы (A–Z);
   - цифры (0–9);
   - специальные символы из набора: ! @ # $ % & * ?.
2. Обеспечить выполнение обязательных требований для каждого сгенерированного пароля:
   - пароль должен содержать хотя бы одну заглавную букву;
   - пароль должен содержать хотя бы одну строчную букву;
   - пароль должен содержать хотя бы одну цифру;
   - пароль должен содержать хотя бы один специальный символ из указанного набора;
   - пароль не должен содержать последовательности из трёх и более одинаковых символов подряд (например, «ааа» или «111» запрещены);

- пароль не должен содержать три и более символа подряд, идущих в алфавитном порядке.

3. После генерации каждого пароля программа должна вывести его на экран вместе с оценкой надёжности:

- «надёжный», если длина пароля 12 и более символов и используются все четыре типа символов;
- «средний», если длина пароля 8–11 символов, но используются все четыре типа;
- «слабый» во всех остальных случаях (включая те, где не выполнились условия — такие пароли генерироваться не должны, но на случай ошибки предусмотрите вывод предупреждения).

6.3. Критерии и шкала оценивания на основе БРС.

Соответствие государственной шкалы оценивания академической успеваемости и шкалы ECTS при зачёте

Оценка по шкале ECTS	**Сумма баллов за все виды учебной деятельности**	**Оценка по государственной шкале**	**Определение**
A	90 – 100	«Отлично»	отличное выполнение с незначительным количеством неточностей
B	80 – 89	«Хорошо»	в целом правильно выполненная работа с незначительным количеством ошибок (до 10%)
C	75 – 79		в целом правильно выполненная работа с незначительным количеством ошибок (до 15%)
D	70 – 74	«Удовлетворительно»	неплохо, но со значительным количеством недостатков
E	60 – 69		выполнение удовлетворяет минимальные критерии
FX	35 – 59	«Не удовлетворительно»	с возможностью повторной сдачи
F	0 – 34		с обязательным повторным изучением дисциплины (выставляется комиссией)

6.4. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий

Компьютер с операционной системой RedOS или Windows с устойчивым Интернет-соединением для работы с ноутбуками Google Colab, программные продукты с открытой лицензией: PyCharm Community Edition, Visual Studio Code, Jupyter Notebook.

7. Методические материалы по освоению дисциплины

Получение углубленных знаний по изучаемой дисциплине достигается за счет дополнительных часов к аудиторной работе самостоятельной работы студентов. Выделяемые часы целесообразно использовать для знакомства с дополнительной научной литературой по проблематике дисциплины, анализа научных концепций и современных подходов к осмыслению рассматриваемых проблем. К самостоятельному виду работы студентов относится работа в библиотеках, в электронных поисковых системах и т.п. по сбору материалов, необходимых для проведения практических занятий или выполнения конкретных заданий преподавателя по изучаемым темам. Студенты могут установить диалог с преподавателем, получать консультации по выполнению заданий. В качестве оценочных средств на протяжении семестра используются практические задания.

Обучение по дисциплине «Информатика и программирование» предполагает изучение курса на аудиторных занятиях (лекции, практические занятия) и самостоятельную работу студентов. Практические занятия дисциплины предполагают их проведение в различных формах с целью выявления полученных знаний, умений, навыков и компетенций с проведением контрольных мероприятий. С целью обеспечения успешного обучения студент должен готовиться к лекции, поскольку она является важнейшей формой организации учебного процесса, поскольку:

– знакомит с новым учебным материалом;
– разъясняет учебные элементы, трудные для понимания;
– систематизирует учебный материал;
– ориентирует в учебном процессе.

Работа обучающегося на лекции:

Слушание и запись лекций – сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность обучающегося. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим обучающимся.

Подготовка к практическим занятиям:

Подготовку к каждому практическому занятию каждый обучающийся должен начать с ознакомления с планом, который отражает содержание предложенной темы. Тщательное продумывание и изучение вопросов плана основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованную к данной теме. Если программой дисциплины предусмотрено выполнение практического задания, то его необходимо выполнить с учетом предложенной инструкции. Все новые понятия по изучаемой теме необходимо внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности обучающегося свободно ответить на теоретические вопросы практического занятия, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий и контрольных работ.

Структура практического занятия:

В зависимости от содержания и количества отведенного времени на изучение каждой темы может практическое занятие состоять из четырех-пяти частей:

1. Устный опрос.
2. Обсуждение теоретических вопросов, определенных программой дисциплины.
3. Выполнение практических заданий с последующим разбором полученных результатов или обсуждение практического задания, выполненного дома.
4. Подведение итогов занятия.

Работа с литературными источниками:

В процессе подготовки к практическим занятиям, обучающимся необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем, что позволяет обучающимся проявить свою индивидуальность в рамках выступления на занятиях, выявить широкий спектр мнений по изучаемой проблеме.

8. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет

8.1. Основная литература

1. PYTHON-программирование : учебное пособие / И. И. Баглаев, Т. Ж. Базаржапова, Н. В. Очирова, Н. В. Юможапова. — Улан-Удэ : БГУ, 2026. — 118 с. — ISBN 978-5-9793-2087-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/519059 (дата обращения: 13.05.2026). — Режим доступа: для авториз. пользователей.

2. Сергеева, О. А. Программирование на Python : учебно-методическое пособие / О. А. Сергеева. — Кемерово : КемГУ, 2024. — 157 с. — ISBN 978-5-8353-3123-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/420758 (дата обращения: 13.05.2026). — Режим доступа: для авториз. пользователей.

8.2. Дополнительная литература

3. Лысаков, К. Ф. Практическое программирование на Python : учебное пособие / К. Ф. Лысаков. — Новосибирск : НГУ, 2023. — 76 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/388274 (дата обращения: 13.05.2026). — Режим доступа: для авториз. пользователей.

4. Василекина, О. М. Учебно-методическое пособие по дисциплине «Алгоритмизация и программирование»: Структурное и процедурное программирование на языке Python направление подготовки 09.03.03 Прикладная информатика профиль «Прикладная информатика в экономике» : учебно-методическое пособие / О. М. Василекина. — Великие Луки : Великолукская ГСХА, 2024. — 104 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/426992 (дата обращения: 13.05.2026). — Режим доступа: для авториз. пользователей.

8.3.Нормативные правовые документы и иная правовая информация

1. Конституция Российской Федерации. – Текст : электронный // Сайт Президента Российской Федерации. – URL: http://www.kremlin.ru/acts/constitution

8.4 Интернет-ресурсы

1. Информационно-правовой портал ГАРАНТ.РУ. – URL: https://www.garant.ru/
2. Научная электронная библиотека eLIBRARY.RU. – URL: https://elibrary.ru/
3. Научная электронная библиотека «КиберЛенинка». – URL: https://cyberleninka.ru

4. Электронно-библиотечная система «Лань». – URL: http://e.lanbook.com

5. Документация по Python – URL: https:// https://docs.python.org/3/

9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

Материально-техническое обеспечение дисциплины включает в себя:

- лекционные аудитории, оборудованные видеопроекционным оборудованием для презентаций, средствами звуковоспроизведения, экраном;
- помещения для проведения практических занятий, оборудованные учебной мебелью.

Дисциплина поддержана соответствующими программными продуктами с открытой лицензией: PyCharm Community Edition, Visual Studio Code, Jupyter Notebook.

Вуз обеспечивает каждого обучающегося рабочим местом в компьютерном классе в соответствии с объемом изучаемых дисциплин, обеспечивает выход в сеть Интернет.

Помещения для самостоятельной работы обучающихся включают следующую оснащенность: столы аудиторные, стулья, доски аудиторные, компьютеры с подключением к локальной сети института (для компьютерных аудиторий) и Интернет. Для изучения учебной дисциплины используются автоматизированная библиотечная информационная система и электронные библиотечные системы.