Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гриб Владислав Валерьевич
Должность: Ректор
Дата подписания: 23.09.2026 19:10:44
Уникальный программный ключ:
637517d24e103c3db032acf37e839d98ec1c5bb2f5eb89c29abfcd7f43985447

Образовательное частное учреждение высшего образования
«МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ А.С. ГРИБОЕДОВА»
(ИМПЭ им. А.С. Грибоедова)

МЕЖДУНАРОДНЫЙ ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И БИЗНЕС-ИНФОРМАТИКИ

УТВЕРЖДАЮ
Директор Международного института информационных технологий и бизнес-информатики
_____ Г.В. Новикова
«17» декабря 2025г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по учебной дисциплине:
ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON
по направлению подготовки/специальности:
09.03.03 Прикладная информатика
(уровень бакалавриат)
профилю/специализации:
«Анализ данных»
Формы обучения: очная, заочная

Москва

Фонд оценочных средств для дисциплины «Язык программирования Python». Направление подготовки/специальность 09.03.03 Прикладная информатика, направленность (профиль/специализация): Анализ данных / О.Ю. Евдокимова – М.: ИМПЭ им. А.С. Грибоедова.

Фонд оценочных средств является неотъемлемой частью рабочей программы дисциплины.

Разработчик: О.Ю. Евдокимова

Заведующий кафедрой: _____/ Н. Н. Загускин/

1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Настоящий Фонд оценочных средств (ФОС) по дисциплине «Язык программирования Python» является неотъемлемым приложением к рабочей программе дисциплины (РПД) «Язык программирования Python». На данный ФОС распространяются все реквизиты утверждения, представленные в РПД по данной дисциплине.

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

Для определения качества освоения обучающимися учебного материала по дисциплине используются следующие оценочные средства:

№ п/п	Оценочное средство	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Тестирование	Вид контроля, позволяющий оценить изученный теоретический материал.	Вопросы для проведения тестирования
2	Практические задания	Вид контроля, позволяющий оценить умение обучающегося применять осваиваемую компетенцию в практических ситуациях и при решении производственных задач	Задания к практическому занятию
3	Контрольная работа	Вид контроля, позволяющий определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций	Задания контрольной работы
4	Самостоятельная работа	Вид контроля, позволяющий оценить проработку теоретического материала, изучение рекомендуемой литературы, выполнение практико-ориентированных заданий (заполнение таблиц, проведение сравнительного анализа, составление схем и др.), решение практических задач, создание презентаций, написание рефератов, подборку нормативного и иного материала и выполнение других заданий	Задания самостоятельной работы
5	Курсовая работа	Вид контроля, позволяющий выявить степень владения базовыми знаниями, умениями и навыками, необходимыми для обучения, и определить уровень владения новым материалом	Индивидуальные задания (темы) для курсовой работы
6	Зачет/Зачет с оценкой/Экзамен	Вид контроля, позволяющий выявить степень овладения знаниями, умениями и навыками, необходимыми для дальнейшего освоения образовательной программы подготовки	Вопросы для подготовки к зачету/зачету с оценкой/экзамену

3. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЕ

3.1. Сопроводительная информация.

Разработчик	О.Ю. Евдокимова
Кафедра	Прикладной информатики и информационных технологий
Наименование дисциплины	Язык программирования Python
Факультет / институт	Институт международной экономики, лидерства и менеджмента
Направление подготовки / специальность	09.03.03 Прикладная информатика
Количество вопросов в оценочных заданиях (диапазон)	От 53 до 106
Общее время тестирования (мин)	90
Общее количество вопросов/заданий в ФОС	106
Размещенность на сайте Университета примерного перечня вопросов, заданий ФОС – для подготовки обучающихся к прохождению оценки (да / нет)	Да

3.2. Характеристика оцениваемых компетенций.

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
ПК-2	Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	**ИПК-2.1 Знать:** Возможности современных и перспективных средств разработки программных продуктов и технических средств **ИПК-2.2 Уметь:** Выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению
ПК-6	Способен применять методы и средства проектирования компьютерного программного обеспечения	**ИПК-6.1 Знать:** Методы и средства проектирования баз данных **ИПК-6.2 Уметь:** Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов
ПК-7	Способен осуществлять проектирование программных интерфейсов	**ИПК-7.1. Знать:** Методы и средства проектирования программных интерфейсов **ИПК-7.2. Уметь:** Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов

4. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ТЕКУЩЕГО КОНТРОЛЯ

4.1. ТИПОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ.

Тесты содержат набор вопросов, в полном объеме охватывающие изученный теоретический материал по указанной теме (индикаторы ЗНАТЬ). Выполнение тестов позволяет определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций. Индивидуальный тестовый сеанс для каждого обучающегося формируется по специальному алгоритму, обеспечивающему заданную тематическую структуру и пропорциональное наличие вопросов разного типа и сложности.

При формировании тестов необходимо использовать задания следующих типов:

Тип задания 1. Задания закрытого типа на установление соответствия.

Тип задания 2. Задания закрытого типа на установление последовательности.

Тип задания 3. Задания комбинированного типа, предполагающие выбор одного правильного ответа из предложенных

Тип задания 4. Задания комбинированного типа, предполагающие выбор нескольких ответов из предложенных

Тип задания 5. Задания открытого типа с развернутым ответом.

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td>Тема 1. Возможности языка программирования Python</td><td>ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению</td><td>ИПК-2.1 Знать: Возможности современных и перспективных средств разработки программных продуктов и технических средств</td><td>1. Задание на установление соответствия (Тип 1)
Соотнесите элементы языка Python с их описаниями:
<table><tr><th>Код</th><th>Элемент</th><th>Код</th><th>Описание</th></tr><tr><td>А</td><td>Переменная</td><td>1</td><td>Функция для вывода информации на экран</td></tr><tr><td>Б</td><td>int</td><td>2</td><td>Именованная область памяти для хранения данных</td></tr><tr><td>В</td><td>input()</td><td>3</td><td>Целочисленный тип данных</td></tr><tr><td>Г</td><td>print()</td><td>4</td><td>Функция для получения данных от пользователя</td></tr></table>
<table><tr><td>А</td><td>Б</td><td>В</td><td>Г</td><td>Д</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>
ОТВЕТ:
<table><tr><td>А</td><td>Б</td><td>В</td><td>Г</td><td>Д</td></tr><tr><td>2</td><td>3</td><td>4</td><td>1</td><td>5</td></tr></table></td></tr>
<tr><td>Тема 1. Возможности языка программирования Python</td><td>ПК-6. Способен осуществлять проектирование баз данных</td><td>ИПК-6.1 Знать: Методы и средства проектирования баз данных</td><td>2.Задание с выбором нескольких ответов (Тип 4)
Какие из следующих типов данных относятся к простым (скалярным) в Python?
1. int
2. list
3. float
4. dict
5. bool
ОТВЕТ: 1, 3, 5</td></tr>
<tr><td>Тема 2. Организация ветвлений.</td><td>ПК-7. Способен осуществлять проектирование программных интерфейсов</td><td>ИПК-7.1. Знать:
Методы и средства проектирования программных интерфейсов</td><td>3. Задание на установление последовательности (Тип 2)
Расположите этапы выполнения инструкции if-elif-else в правильной последовательности:
1. Выполнение блока кода после первого условия, которое истинно
2. Проверка условия после if
3. Проверка условий после elif (если есть)</td></tr>
</table>

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			4. Выполнение блока после else (если все условия ложны) 5. Завершение условной конструкции | | | | | | | ОТВЕТ: | 2 | 3 | 1 | 4 | 5 |
Тема 2. Организация ветвлений.	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	ИПК-2.1 Знать: Возможности современных и перспективных средств разработки программных продуктов и технических средств	4. Задание с выбором одного ответа(Тип 3) Что выведет код: print(5 > 3 and 2 < 4 or 1 == 0) 1. True 2. False 3. Ошибку 4. None ОТВЕТ: 1) True — потому что 5 > 3 истинно, 2 < 4 истинно, оператор and имеет higher precedence, чем or
Тема 1.3 Организация повторений	ПК-7. Способен осуществлять проектирование программных интерфейсов	ИПК-7.1. Знать: Методы и средства проектирования программных интерфейсов	5. Задание с выбором одного ответа(Тип 3) Чем принципиально отличается цикл for от цикла while в Python? 1. for используется только для чисел, while — для строк 2. for выполняет фиксированное число итераций, while — пока условие истинно 3. for может быть бесконечным, while — всегда конечный 4. while работает быстрее, чем for ОТВЕТ: 2) for выполняет фиксированное число итераций, while — пока условие истинно — цикл for итерируется по коллекции, а while выполняется до изменения условия
Тема 3. Организация повторений.	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.1 Знать: Методы и средства проектирования баз данных	6. Задание на установление последовательности (Тип 2) Расположите этапы выполнения цикла for item in collection: в правильной последовательности: 1. Выполнение тела цикла для текущего элемента 2. Переход к следующему элементу коллекции 3. Инициализация итератора 4. Проверка наличия следующего элемента 5. Завершение цикла при исчерпании элементов | | | | | | |

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td></td><td></td><td></td><td>ОТВЕТ: 3 → 4 → 1 → 2 → 4 → 5</td></tr>
<tr><td>Тема 3. Организация повторений.</td><td>ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению</td><td>ИПК-2.1 Знать: Возможности современных и перспективных средств разработки программных продуктов и технических средств</td><td>7.Задание на установление соответствия (Тип 1)
Соотнесите циклические конструкции с их описаниями:<table><tr><th>Код</th><th>Конструкция</th><th>Код</th><th>Описание</th></tr><tr><td>А</td><td>for</td><td>1</td><td>Выполняется пока условие истинно</td></tr><tr><td>Б</td><td>while</td><td>2</td><td>Итерируется по элементам коллекции</td></tr><tr><td>В</td><td>break</td><td>3</td><td>Досрочно прерывает цикл</td></tr><tr><td>Г</td><td>continue</td><td>4</td><td>Переходит к следующей итерации цикла</td></tr><tr><td>Д</td><td>else в цикле</td><td>5</td><td>Выполняется если цикл завершился нормально</td></tr></table><table><tr><td></td><td></td><td></td><td></td><td></td></tr></table>
ОТВЕТ: А-2, Б-1, В-3, Г-4, Д-5</td></tr>
<tr><td>Тема 4. Упорядоченные типы данных.</td><td>ПК-7. Способен осуществлять проектирование программных интерфейсов</td><td>ИПК-7.1. Знать:
Методы и средства проектирования программных интерфейсов</td><td>8. Задание с выбором нескольких ответов (Тип 4)
Какие из следующих операций поддерживаются и строками, и списками в Python?
1. Срезы (slicing)
2. Конкатенация через +
3. Изменение элемента по индексу
4. Метод append()
5. Метод find()
ОТВЕТ: 1, 2</td></tr>
<tr><td>Тема 4. Упорядоченные типы данных.</td><td>ПК-6. Способен осуществлять проектирование баз данных</td><td>ИПК-6.1 Знать: Методы и средства проектирования баз данных</td><td>9. Задание открытого типа (Тип 5)
Опишите различия между списками (list) и кортежами (tuple) в Python. В каких случаях предпочтительнее использовать каждый из этих типов данных?
ОТВЕТ:</td></tr>
</table>

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td></td><td></td><td></td><td>• Списки: изменяемые (mutable), поддерживают добавление, удаление, изменение элементов. Используются для хранения однородных данных, которые могут изменяться
• Кортежи: неизменяемые (immutable), занимают меньше памяти, работают быстрее. Используются для гетерогенных данных, которые не должны изменяться (координаты, записи БД)
• Пример: Список для студентов в группе (может меняться), кортеж для координат точки (фиксированно)</td></tr>
<tr><td>Тема 4. Упорядоченные типы данных.</td><td>ПК-7. Способен осуществлять проектирование программных интерфейсов</td><td>ИПК-7.1. Знать:
Методы и средства проектирования программных интерфейсов</td><td>10. Задание с выбором нескольких ответов (Тип 4)
Какие методы списков изменяют исходный список?
1. append()
2. sort()
3. sorted()
4. reverse()
5. len()
ОТВЕТ: 1, 2, 4</td></tr>
<tr><td>Тема 4. Упорядоченные типы данных.</td><td>ПК-6. Способен осуществлять проектирование баз данных</td><td>ИПК-6.1 Знать: Методы и средства проектирования баз данных</td><td>11.Задание с выбором одного ответа(Тип 3)
Что такое срез (slicing) в Python?
1. Удаление элементов из коллекции
2. Получение подпоследовательности по индексам [start:stop:step]
3. Сортировка коллекции
4. Объединение двух коллекций
ОТВЕТ: 2) Получение подпоследовательности по индексам [start:stop:step] — позволяет извлекать части строк, списков, кортежей</td></tr>
<tr><td>Тема 5. Неупорядоченные типы данных.</td><td>ПК-6. Способен осуществлять проектирование баз данных</td><td>ИПК-6.1 Знать: Методы и средства проектирования баз данных</td><td>12. Задание на установление соответствия (Тип 1)
Соотнесите типы данных с их характеристиками:
<table>
<tr><th>Код</th><th>Тип данных</th><th>Код</th><th>Характеристика</th></tr>
<tr><td>А</td><td>set</td><td>1</td><td>Коллекция пар "ключ-значение"</td></tr>
<tr><td>Б</td><td>dict</td><td>2</td><td>Неупорядоченная коллекция уникальных элементов</td></tr>
<tr><td>В</td><td>list</td><td>3</td><td>Изменяемая упорядоченная коллекция</td></tr>
<tr><td>Г</td><td>tuple</td><td>4</td><td>Неизменяемая упорядоченная коллекция</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td></td><td></td><td></td><td>ОТВЕТ:<table><tr><td>А</td><td>Б</td><td>В</td><td>Г</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>
Правильный ответ<table><tr><td>А</td><td>Б</td><td>В</td><td>Г</td></tr><tr><td>2</td><td>1</td><td>3</td><td>4</td></tr></table></td></tr>
<tr><td>Тема 5. Неупорядоченные типы данных.</td><td>ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению</td><td>ИПК-2.1 Знать: Возможности современных и перспективных средств разработки программных продуктов и технических средств</td><td>13. Задание с выбором одного ответа(Тип 3)
Какой метод словаря в Python возвращает значение по ключу, если ключ существует, или значение по умолчанию, если ключ отсутствует?
1. get()
2. setdefault()
3. update()
4. pop()
ОТВЕТ: 1) get() — позволяет безопасно получить значение без риска возникновения KeyError</td></tr>
<tr><td>Тема 5. Неупорядоченные типы данных.</td><td>ПК-6. Способен осуществлять проектирование баз данных</td><td>ИПК-6.1 Знать: Методы и средства проектирования баз данных</td><td>14.Задание с выбором нескольких ответов (Тип 4)
Какие операции поддерживают множества (set) в Python?
1. add() — добавление элемента
2. remove() — удаление элемента
3. keys() — получение ключей
4. union() — объединение
5. intersection() — пересечение
ОТВЕТ: 1, 2, 4, 5</td></tr>
<tr><td>Тема 5. Неупорядоченные типы данных.</td><td>ПК-7. Способен осуществлять проектирование программных интерфейсов</td><td>ИПК-7.1. Знать:
Методы и средства проектирования программных интерфейсов</td><td>15.Задание на установление соответствия (Тип 1)
Соотнесите операции с множествами с их результатами:<table><tr><td>Код</td><td>Операция</td><td>Код</td><td>Результат для {1,2,3} и {2,3,4}</td></tr><tr><td>А</td><td>union()</td><td>1</td><td>{2, 3}</td></tr><tr><td>Б</td><td>intersection()</td><td>2</td><td>{1, 2, 3, 4}</td></tr></table></td></tr>
</table>

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>В</td><td>difference()</td><td>3</td><td>{1}</td></tr>
<tr><td>Г</td><td>symmetric_difference()</td><td>4</td><td>{1, 4}</td></tr>
</table>
ОТВЕТ: А-2, Б-1, В-3, Г-4
</td></tr>
<tr><td>Тема 6. Вложенные структуры данных.</td><td>ПК-6. Способен осуществлять проектирование баз данных</td><td>ИПК-6.1 Знать: Методы и средства проектирования баз данных</td><td>16. Задание с выбором нескольких ответов (Тип 4)
Какие из следующих структур являются корректными вложенными структурами данных в Python?
1. list = [1, 2, [3, 4]]
2. dict = {'key': {'nested_key': 'value'}}
3. set = {{1, 2}, {3, 4}}
4. tuple = (1, (2, 3), 4)
5. list = [[1, 2], [3, 4]]
ОТВЕТ: 1, 2, 4, 5</td></tr>
<tr><td>Тема 6. Вложенные структуры данных.</td><td>ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению</td><td>ИПК-2.1 Знать: Возможности современных и перспективных средств разработки программных продуктов и технических средств</td><td>17. Задание открытого типа (Тип 5)
Опишите, как получить доступ к элементу со значением 'target' в структуре data = {'level1': {'level2': ['a', 'b', 'target']}}. Какие операции индексации необходимы?
ОТВЕТ:
python
data['level1']['level2'][2]
• Первый индекс ['level1'] обращается к значению первого словаря
• Второй индекс ['level2'] обращается к вложенному словарю
• Третий индекс [2] обращается к третьему элементу списка
ОТВЕТ:3) Переменные, указанные в объявлении функции - они определяют тип и количество принимаемых аргументов</td></tr>
<tr><td>Тема 7. Функции и модули.</td><td>ПК-7. Способен осуществлять проектирование программных интерфейсов</td><td>ИПК-7.1. Знать:
Методы и средства проектирования программных интерфейсов</td><td>18. Задание на установление последовательности (Тип 2)
Расположите этапы работы с функцией в Python в правильной последовательности:
1. Передача аргументов в параметры функции
2. Определение функции с помощью def
3. Возврат результата с помощью return
4. Вызов функции по имени
5. Выполнение тела функции
<table>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			ОТВЕТ: 2 → 4 → 1 → 5 → 3
Тема 7. Функции и модули.	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	ИПК-2.1 Знать: Возможности современных и перспективных средств разработки программных продуктов и технических средств	19. Задание с выбором одного ответа(Тип 3) Что такое область видимости (scope) в Python? 1. Размер файла с кодом 2. Часть программы, в которой имя ссылается на определенный объект 3. Количество строк в функции 4. Тип возвращаемого значения функции ОТВЕТ: 2) Часть программы, в которой имя ссылается на определенный объект — определяет, где переменная может быть использована
Тема 7. Функции и модули.	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.1 Знать: Методы и средства проектирования баз данных	20.Задание с выбором одного ответа(Тип 3) Что такое параметр по умолчанию в функции Python? 1. Параметр, который нельзя изменить 2. Параметр, который имеет значение, используемое если аргумент не передан 3. Параметр, который всегда должен быть последним 4. Параметр, который принимает только числа ОТВЕТ: 2) Параметр, который имеет значение, используемое если аргумент не передан — позволяет вызывать функцию с меньшим количеством аргументов
Тема 7. Функции и модули.	ПК-7. Способен осуществлять проектирование программных интерфейсов	ИПК-7.1. Знать: Методы и средства проектирования программных интерфейсов	21.Задание на установление последовательности (Тип 2) Расположите этапы импорта модуля в Python в правильной последовательности: 1. Поиск модуля в системных путях 2. Использование функций/классов из модуля 3. Выполнение кода в импортируемом модуле 4. Импорт модуля с помощью import 5. Создание пространства имен модуля | | | | | | ОТВЕТ: 4 → 1 → 3 → 5 → 2
Тема 8. Файлы.	ПК-6. Способен осуществлять проектирование баз	ИПК-6.1 Знать: Методы и средства проектирования баз данных	22. Задание на установление соответствия (Тип 1) Соотнесите методы работы с файлами с их описаниями:

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td></td><td>данных</td><td></td><td>
<table>
<tr><th>Код</th><th>Метод</th><th>Код</th><th>Описание</th></tr>
<tr><td>А</td><td>open()</td><td>1</td><td>Читает одну строку из файла</td></tr>
<tr><td>Б</td><td>read()</td><td>2</td><td>Открывает файл и возвращает файловый объект</td></tr>
<tr><td>В</td><td>readline()</td><td>3</td><td>Читает все содержимое файла</td></tr>
<tr><td>Г</td><td>write()</td><td>4</td><td>Записывает данные в файл</td></tr>
<tr><td>Д</td><td>close()</td><td>5</td><td>Закрывает файл и освобождает ресурсы</td></tr>
</table>
<table>
<tr><th>А</th><th>Б</th><th>В</th><th>Г</th><th>Д</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>
Ответ:
<table>
<tr><th>А</th><th>Б</th><th>В</th><th>Г</th><th>Д</th></tr>
<tr><td>2</td><td>3</td><td>1</td><td>4</td><td>5</td></tr>
</table>
</td></tr>
<tr><td>Тема 8. Файлы.</td><td>ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению</td><td>ИПК-2.1 Знать: Возможности современных и перспективных средств разработки программных продуктов и технических средств</td><td>23. Задание с выбором нескольких ответов (Тип 4)
Какие из следующих утверждений о работе с файлами в Python верны?
1. Файл нужно обязательно закрывать после работы
2. Контекстный менеджер with автоматически закрывает файл
3. Файл можно открывать одновременно на чтение и запись
4. Метод read() всегда возвращает список строк
5. Файлы бывают текстовые и бинарные
ОТВЕТ: 2, 3, 5</td></tr>
</table>

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
	ПК-7. Способен осуществлять проектирование программных интерфейсов	ИПК-7.1. Знать: Методы и средства проектирования программных интерфейсов	24.Задание на установление последовательности (Тип 2) Расположите этапы чтения данных из файла в правильной последовательности: 1. Открытие файла в режиме чтения 2. Обработка прочитанных данных 3. Чтение содержимого файла 4. Закрытие файла 5. Проверка существования файла ОТВЕТ: 5 → 1 → 3 → 2 → 4
Тема 8. Файлы.	ПК-7. Способен осуществлять проектирование программных интерфейсов	ИПК-7.1. Знать: Методы и средства проектирования программных интерфейсов	25.Задание с выбором одного ответа(Тип 3) Для чего используется режим 'a' при открытии файла? 1. Для чтения файла 2. Для записи с очисткой файла 3. Для добавления данных в конец файла 4. Для чтения и записи одновременно ОТВЕТ: 3) Для добавления данных в конец файла — режим 'append' не удаляет существующее содержимое

Критерии оценивания тестового задания:

Оценка	Критерии оценивания
отлично	от 90 до 100 % правильно выполненных заданий
хорошо	от 70 до 89 % правильно выполненных заданий
удовлетворительно	от 50 до 69 % правильно выполненных заданий
неудовлетворительно	менее 50 % правильно выполненных заданий

4.2 ТИПОВЫЕ ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Практические задания должны отражать умение обучающегося применять осваиваемую компетенцию в практических ситуациях и при решении производственных задач (индикаторы УМЕТЬ).

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
Тема 1. Возможности языка программирования Python	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных ин-терфейсов	1. Программа работы с переменными и типами данных python def demo_variables(): a,b,c,d=10,3.14,"Hello",True print(f"int:{a}, float:{b}, str:{c}, bool:{d}") print(f"Types: {type(a)},{type(b)},{type(c)},{type(d)}") e=str(a)+c; print(f"Конкатенация: {e}") f=float(input("Введите число: ")); print(f"Введено: {f}") demo_variables()
Тема 2. Организация ветвлений.			2. Программа с операторами сравнения и ветвлениями python def demo_branching(): x,y=10,5; print(f"Числа: {x},{y}") if x>y: print(f"{x} > {y}") elif x<y: print(f"{x} < {y}") else: print(f"{x} = {y}") age=int(input("Возраст: ")) status="взрослый" if age>=18 else "ребенок"; print(f"Статус: {status}") demo_branching()
Тема 3. Организация повторений.	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных ин-терфейсов	3. Программа с циклами for и while python def demo_loops(): print("Цикл for:"); [print(i) for i in range(1,4)] print("Цикл while:"); n=1 while n<=3: print(n); n+=1 nums=[2,4,6]; print("For по списку:") for num in nums: print(f"Квадрат {num}: {num**2}") demo_loops()
Тема 4. Упорядоченные типы данных.	ПК-6. Способен осуществлять	ИПК-6.2 Уметь: Применять методы и средства	4. Программа работы со строками и списками python

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
	проектирование баз данных	проектирования компьютерного программного обеспечения, структур данных, баз данных, программных ин-терфейсов	def demo_ordered_types(): s="Python"; print(f"Строка: {s}") print(f"Срезы: [1:4]='{s[1:4]}', [::2]='{s[::2]}'") lst=[1,2,3]; print(f"Список: {lst}") lst.append(4); lst.insert(0,0); print(f"После изменений: {lst}") print(f"Длина: {len(s)}, {len(lst)}") demo_ordered_types()
Тема 4. Упорядоченные типы данных.	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	ИПК-2.2 Уметь: Выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению	5. Программа работы с кортежами python def demo_tuples(): t=(1,2,3); print(f"Кортеж: {t}") a,b,c=t; print(f"Распаковка: a={a}, b={b}, c={c}") lst=list(t); lst.append(4); t_new=tuple(lst) print(f"Преобразование: {t_new}") coord=(10,20); x,y=coord; print(f"Координаты: x={x}, y={y}") demo_tuples()
Тема 5. Неупорядоченные типы данных.	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	6. Программа работы с множествами python def demo_sets(): set1={1,2,2,3}; set2={3,4,5}; print(f"Множества: {set1}, {set2}") print(f"Объединение: {set1|set2}") print(f"Пересечение: {set1&set2}") print(f"Разность: {set1-set2}") set1.add(4); print(f"После add: {set1}") demo_sets()
Тема 5. Неупорядоченные типы данных.	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	7. Программа работы со словарями python def demo_dicts(): student={"name":"Иван","age":20,"grades":[5,4,3]} print(f"Студент: {student}") print(f"Имя: {student['name']}, Возраст: {student.get('age')}") student["city"]="Москва"; print(f"После добавления: {student}") for key,val in student.items(): print(f"{key}: {val}") demo_dicts()

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию

Тема 6. Вложенные структуры данных.	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	7. Программа работы с вложенными структурами python
			def demo_nested_structures():
			students=[{"name":"Иван","grades":{"math":5,"physics":4}},
			{"name":"Мария","grades":{"math":4,"physics":5}}]
			for student in students:
			name=student["name"]; math_grade=student["grades"]["math"]
			print(f"{name}: математика - {math_grade}")
			matrix=[[1,2],[3,4]]; print(f"Матрица: {matrix}")
			print(f"Элемент [1][0]: {matrix[1][0]}")
			demo_nested_structures()

Тема 7. Функции и модули.	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	8. Программа с пользовательскими функциями python def greet(name): return f"Привет, {name}!" def calculate(a,b): return a+b,a*b def functions_demo(): print(greet("Анна")) sum_result,mult_result=calculate(3,4) print(f"Сумма: {sum_result}, Произведение: {mult_result}") numbers=[1,2,3]; squares=list(map(lambda x:x**2,numbers)) print(f"Квадраты: {squares}") functions_demo()
Тема 8. Файлы.	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных ин-терфейсов	9 Программа работы с файлами python def file_operations(): with open("data.txt","w") as f: f.write("Строка1\nСтрока2\n") with open("data.txt","r") as f: lines=f.readlines() print("Содержимое файла:"); [print(line.strip()) for line in lines] with open("data.txt","a") as f: f.write("Строка3\n") file_operations()
Тема 2. Организация ветвлений.	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных,	10. Программа "Классификатор чисел" python def number_classifier(): num=int(input("Введите число: ")) if num>0: print("Положительное")

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
		программных интерфейсов	elif num<0: print("Отрицательное") else: print("Ноль") if num%2==0: print("Четное"); return print("Нечетное") number_classifier()
Тема 3. Организация повторений.	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	ИПК-2.2 Уметь: Выявлять взаимосвязи и документировать требо-вания к компьютерному программному обеспе-чению	11. Программа "Таблица умножения" python def multiplication_table(): n=int(input("Число: ")); print(f"Таблица умножения для {n}:") for i in range(1,11): print(f"{n} × {i} = {n*i}") print("Квадраты 1-5:"); squares=[i**2 for i in range(1,6)] print(squares); print("Четные 1-10:"); evens=[i for i in range(2,11,2)] print(evens) multiplication_table()
Тема 4. Упорядоченные типы данных.	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	12. Программа "Анализатор текста" python def text_analyzer(): text=input("Введите текст: ").lower(); words=text.split() print(f"Слов: {len(words)}, Символов: {len(text)}") print(f"Первое слово: '{words[0]}', Последнее: '{words[-1]}'") unique_words=set(words); print(f"Уникальных слов: {len(unique_words)}") print(f"Реверс текста: '{text[::-1]}'") text_analyzer()
Тема 5. Неупорядоченные типы данных.	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	13. Программа "Управление коллекциями" python def collection_manager(): # *Множество уникальных элементов* unique_nums={1,2,3,3,4}; print(f"Уникальные: {unique_nums}") # *Словарь с данными* person={"name":"Анна","age":25,"city":"СПб"} print(f"Данные: {person}"); person["email"]="anna@mail.ru" print(f"После добавления: {person}") # *Операции с множествами*

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			set_a={1,2,3}; set_b={3,4,5}; print(f"Объединение: {set_a|set_b}") collection_manager()
Тема 6. Вложенные структуры данных.	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных ин-терфейсов	14. Программа "Учебный журнал" python def study_journal(): journal={"Математика":[4,5,3],"Физика":[5,5,4]} print("Журнал оценок:"); [print(f"{subj}: {grades}") for subj,grades in journal.items()] math_avg=sum(journal["Математика"])/len(journal["Математика"]) print(f"Средний по математике: {math_avg:.1f}") journal["Информатика"]=[5,4,5]; print("После добавления:"); print(journal) study_journal()
Тема 7. Функции и модули.	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	ИПК-2.2 Уметь: Выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению	15. Программа "Математический модуль" python import math def math_module_demo(): print(f"√16={math.sqrt(16)}, 2³={math.pow(2,3)}") print(f"π={math.pi:.2f}, sin(90)={math.sin(math.pi/2):.1f}") nums=[2.7,2.3,2.5]; print(f"Округление: {[math.ceil(n) for n in nums]}") print(f"Факториал 5={math.factorial(5)}") print(f"НОД(12,18)={math.gcd(12,18)}") math_module_demo()
Тема 8. Файлы.	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	16. Программа "Дневник настроения" python def mood_diary(): mood=input("Ваше настроение: "); date="2024-10-25" with open("diary.txt","a",encoding="utf-8") as f: f.write(f"{date}: {mood}\n") with open("diary.txt","r",encoding="utf-8") as f: print("История настроений:"); [print(line.strip()) for line in f.readlines()[-3:]]

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			mood_diary()
Комплексное задание	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	ИПК-2.2 Уметь: Выявлять взаимосвязи и документировать требо-вания к компьютерному программному обеспе-чению	17. Программа "Учет студентов" python def student_tracker(): students=[{"name":"Иван","age":20,"subjects":["Математика","Физика"]}, {"name":"Мария","age":19,"subjects":["Информатика"]}] print("Список студентов:"); [print(s["name"]) for s in students] adult_students=[s for s in students if s["age"]>=20] print(f"Совершеннолетние: {len(adult_students)}") all_subjects=set(); [all_subjects.update(s["subjects"]) for s in students] print(f"Все предметы: {all_subjects}") student_tracker()
Итоговое задание	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	17. Программа "Мини-база данных" python def mini_database(): data=[]; data.append({"name":"Товар1","price":100,"quantity":5}) data.append({"name":"Товар2","price":200,"quantity":3}) print("База товаров:"); [print(f"{item['name']} - {item['price']} руб.") for item in data] total_value=sum(item["price"]*item["quantity"] for item in data) print(f"Общая стоимость: {total_value} руб.") expensive_items=[item for item in data if item["price"]>150] print(f"Дорогие товары: {[item['name'] for item in expensive_items]}") mini_database()

Критерии оценивания практических занятий:

Оценка	Критерии оценивания
отлично	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи, формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, обосновывает свои суждения и даёт правильные ответы на вопросы преподавателя

хорошо	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи и формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, но содержание ответов имеют некоторые неточности и требуют уточнения и комментария со стороны преподавателя
удовлетворительно	Выставляется, если обучающийся знает и понимает материал по заданной теме, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, студент не может обосновать свои ответы на уточняющие вопросы преподавателя
неудовлетворительно	Выставляется, если обучающийся допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал. Делает ошибки в ответах на уточняющие вопросы преподавателя

4.3. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ КОНТРОЛЬНЫХ РАБОТ

Контрольные работы содержат несколько практических заданий по индивидуальным вариантам, в полном объеме охватывающих изученный материал по указанной теме (индикаторы УМЕТЬ). Выполнение контрольных работ позволяет определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
Тема 2. Организация ветвлений.	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	ИПК-2.2 Уметь: Выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению	Вариант 1 python # 1. Условный оператор: Проверка возраста age = int(input("Введите возраст: ")) if age >= 18: print("Совершеннолетний") else: print("Несовершеннолетний") # 2. Логические операторы: Проверка диапазона num = int(input("Введите число: ")) if 10 <= num <= 50: print("В диапазоне 10-50") else: print("Вне диапазона") # 3. Вложенные условия: Классификация оценок grade = int(input("Оценка: ")) if grade == 5: print("Отлично") elif grade == 4: print("Хорошо") elif grade == 3: print("Удовлетворительно") else: print("Неудовлетворительно")
Тема 3 Организация повторений	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных	Вариант 2 python # 1. Цикл for: Сумма четных чисел total = 0

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
		интерфейсов	for i in range(2, 101, 2): total += i print(f"Сумма четных 2-100: {total}") # 2. Цикл while: Обратный отсчет n = 5 while n > 0: print(n); n -= 1 print("Старт!") # 3. Вложенные циклы: Таблица умножения for i in range(1, 4): for j in range(1, 4): print(f"{i}×{j}={i*j}", end=" ") print()
Тема 4 Упорядоченные типы данных	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Вариант 3 python # 1. Работа со списком: Поиск максимального numbers = [3, 7, 2, 9, 1] max_val = max(numbers) print(f"Максимальный элемент: {max_val}") # 2. Срезы списка: Обратный порядок lst = [1, 2, 3, 4, 5] reversed_lst = lst[::-1] print(f"Обратный список: {reversed_lst}") # 3. Методы строк: Анализ текста text = "Программирование на Python" print(f"Длина: {len(text)}") print(f"Верхний регистр: {text.upper()}") print(f"Замена: {text.replace('Python', 'Java')}")
Тема 5 Неупорядоченные типы данных	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Вариант 4 python # 1. Работа со словарем: Данные студента student = {"name": "Иван", "age": 20, "group": "ПИ-101"} print(f"Студент: {student['name']}") student["gpa"] = 4.5 print(f"Обновленные данные: {student}")

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			# 2. Операции с множествами: Уникальные элементы set1 = {1, 2, 3, 3, 4} set2 = {3, 4, 5, 6} print(f"Уникальные: {set1}") print(f"Пересечение: {set1 & set2}") # 3. Словарные методы: Перебор элементов for key, value in student.items(): print(f"{key}: {value}")

Тема 6 Вложенные структуры данных	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	ИПК-2.2 Уметь: Выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению	Вариант 5
			python
			# 1. Вложенные списки: Матрица
			matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
			print(f"Матрица 2x2: {[row[:2] for row in matrix[:2]]}")
			
			# 2. Вложенные словари: База данных
			library = {
			  "book1": {"title": "Python", "year": 2020},
			  "book2": {"title": "Java", "year": 2019}
			}
			print(f"Название книги: {library['book1']['title']}")
			
			# 3. Доступ к вложенным элементам
			grades = {"math": [5, 4, 3], "physics": [4, 5]}
			print(f"Оценка по математике: {grades['math'][0]}")

Тема 7 Функции и модули	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Вариант 6 python # 1. Простая функция: Калькулятор def multiply(a, b): return a * b result = multiply(5, 3) print(f"Произведение: {result}") # 2. Функция с параметром по умолчанию def greet(name="Гость"): return f"Привет, {name}!" print(greet()); print(greet("Анна")) # 3. Использование модуля math

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			import math print(f"Квадратный корень: {math.sqrt(25)}") print(f"Округление вверх: {math.ceil(4.3)}")

Тема 8. Файлы.	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению		**Вариант 7**
			python
			# 1. Запись в файл
			with open("output.txt", "w") as f:
			  f.write("Первая строка\nВторая строка\n")
			print("Данные записаны")
			
			# 2. Чтение из файла
			with open("output.txt", "r") as f:
			  lines = f.readlines()
			print(f"Прочитано строк: {len(lines)}")
			
			# 3. Обработка данных из файла
			with open("output.txt", "r") as f:
			  for i, line in enumerate(f, 1):
			  print(f"Строка {i}: {line.strip()}")

Комплексное задание	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению
	ПК-6. Способен осуществлять проектирование баз данных
	ПК-7 Способен осуществлять проектирование программных интерфейсов
		ИПК-2.2 Уметь: Выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению
		ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов
		ИПК-7.2. Уметь:
		Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов
			Вариант 8
			python
			# 1. Обработка пользовательского ввода
			name = input("Введите имя: ")
			age = int(input("Введите возраст: "))
			print(f"Данные: {name}, {age} лет")
			
			# 2. Работа с коллекциями
			numbers = [int(x) for x in input("Введите числа через пробел: ").split()]
			print(f"Сумма: {sum(numbers)}")
			
			# 3. Условная логика и циклы
			for num in numbers:
			  if num % 2 == 0: print(f"{num} - четное")
			  else: print(f"{num} - нечетное")

Комбинированно	ПК-2. Способен	ИПК-2.2 Уметь: Выявлять взаимосвязи и	Вариант 9

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
е задание	осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению ПК-6. Способен осуществлять проектирование баз данных ПК-7 Способен осуществлять проектирование программных интерфейсов	документировать требования к компьютерному программному обеспечению ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	python # 1. Создание и модификация списка fruits = ["apple", "banana", "cherry"] fruits.append("orange") fruits.remove("banana") print(f"Фрукты: {fruits}") # 2. Работа со множествами unique_fruits = set(fruits) unique_fruits.add("grape") print(f"Уникальные фрукты: {unique_fruits}") # 3. Преобразование типов numbers_str = ["1", "2", "3"] numbers_int = [int(x) for x in numbers_str] print(f"Числа: {numbers_int}")

Интегрированно е задание	ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению
	ПК-6. Способен осуществлять проектирование баз данных
	ПК-7 Способен осуществлять проектирование программных интерфейсов
		ИПК-2.2 Уметь: Выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению
		ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов
		ИПК-7.2. Уметь:
		Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов
			Вариант 10
			python
			# 1. Функция для работы со словарем
			def get_student_info():
			  return {"name": "Мария", "subjects": ["math", "physics"]}
			
			student = get_student_info()
			print(f"Студент: {student['name']}")
			
			# 2. Обработка вложенных данных
			subjects = student["subjects"]
			print(f"Предметы: {', '.join(subjects)}")
			
			# 3. Запись результатов в файл
			with open("student.txt", "w") as f:
			  f.write(f"Имя: {student['name']}\n")
			  f.write(f"Кол-во предметов: {len(subjects)}\n")
			print("Информация сохранена")

Критерии оценивания контрольной работы:

Оценка	Критерии оценивания
отлично	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи, формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, обосновывает свои суждения и даёт правильные ответы на вопросы преподавателя
хорошо	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи и формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, но содержание ответов имеют некоторые неточности и требуют уточнения и комментария со стороны преподавателя.
удовлетворительно	Выставляется, если обучающийся знает и понимает материал по заданной теме, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, студент не может обосновать свои ответы на уточняющие вопросы преподавателя
неудовлетворительно	Выставляется, если обучающийся допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал. Делает ошибки в ответах на уточняющие вопросы преподавателя

4.4. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа (индикаторы ЗНАТЬ, УМЕТЬ – на выбор) включает в себя проработку теоретического материала, изучение рекомендуемой литературы, выполнение практико-ориентированных заданий (заполнение таблиц, проведение сравнительного анализа, составление схем и др.), решение практических задач, создание презентаций, написание рефератов, подборка нормативного и иного материала и выполнение других заданий.

<table>
<tr><th>Тема</th><th>Код компетенции</th><th>Индикаторы достижения компетенции</th><th>Типовое задание и решение</th></tr>
<tr><td>Тема 1
Возможности языка Python</td><td>ПК-2. Способен осуществлять оценку времени и трудоёмкости реализации требований к ПО</td><td>ИПК-2.1 Знать: Возможности средств разработки
ИПК-2.2 Уметь: Выявлять взаимосвязи требований к ПО</td><td>1. Установите соответствие между типами данных в Python и их характеристиками
Соотнесите каждый тип данных с его описанием и особенностями:
<table>
<tr><th>Пункт</th><th>Тип данных</th><th>Пункт</th><th>Описание</th></tr>
<tr><td>А</td><td>int</td><td>3</td><td>Целочисленный тип произвольной точности</td></tr>
<tr><td>Б</td><td>float</td><td>5</td><td>Вещественный тип двойной точности</td></tr>
<tr><td>В</td><td>str</td><td>7</td><td>Неизменяемая упорядоченная последовательность символов</td></tr>
<tr><td>Г</td><td>list</td><td>6</td><td>Изменяемая упорядоченная последовательность</td></tr>
<tr><td>Д</td><td>tuple</td><td>1</td><td>Неизменяемая последовательность элементов</td></tr>
<tr><td>Е</td><td>dict</td><td>4</td><td>Коллекция пар ключ-значение с быстрым доступом</td></tr>
<tr><td>Ж</td><td>set</td><td>2</td><td>Изменяемая неупорядоченная коллекция уникальных элементов</td></tr>
<tr><td>З</td><td>bool</td><td>8</td><td>Логический тип (True/False)</td></tr>
</table>
2. Исследование основных конструкций Python
Напишите программу, которая демонстрирует работу с переменными и базовыми операциями:
python
name = input("Введите ваше имя: ")
age = int(input("Введите ваш возраст: "))
print(f"Привет, {name}! Тебе {age} лет.")
print(f"Через 5 лет тебе будет {age + 5} лет.")
print(f"Тип переменной name: {type(name)}, тип переменной age: {type(age)}")
3. Анализ преобразования типов данных
Исследуйте и продемонстрируйте преобразование между различными типами данных:
python
number_str = "123"
number_int = int(number_str)
number_float = float(number_str)
print(f"Исходная строка: '{number_str}', тип: {type(number_str)}")</td></tr>
</table>

Тема	Код компетенции	Индикаторы достижения компетенции	Типовое задание и решение
			print(f"Преобразование в int: {number_int}, тип: {type(number_int)}") print(f"Преобразование в float: {number_float}, тип: {type(number_float)}") print(f"Обратное преобразование: {str(number_int)}, тип: {type(str(number_int))}")

Тема 2 Организация ветвлений	ПК-6. Способен осуществлять проектирование баз данных	ИПК-6.1 Знать: Методы проектирования БД
		ИПК-6.2 Уметь: Применять методы проектирования
			1. Составьте блок-схемы алгоритмов
			Разработайте блок-схемы для следующих алгоритмов:
			• Определение знака числа:
			python
			def check_number_sign(num):
			if num > 0:
			return "Положительное"
			elif num < 0:
			return "Отрицательное"
			else:
			return "Ноль"
			• Классификация возраста:
			python
			def classify_age(age):
			if age < 18:
			return "Ребенок"
			elif age < 65:
			return "Взрослый"
			else:
			return "Пенсионер"
			2. Проанализируйте эффективность различных условных конструкций
			Заполните таблицу сравнения условных конструкций в Python:
			• if-elif-else: для множественных условий - Гибкость, читаемость - Может быть громоздкой
			• Вложенный if: для зависимых условий - Логическая группировка - Сложность отладки

Тема	Код компетенции	Индикаторы достижения компетенции	Типовое задание и решение

			• Тернарный оператор: для простых условий - Компактность - Ограниченная функциональность
			3. Исследование логических операторов
			Проведите сравнительный анализ логических операторов:
			python
			a, b, c = True, False, True
			print(f"a and b: {a and b}") # Логическое И
			print(f"a or b: {a or b}") # Логическое ИЛИ
			print(f"not a: {not a}") # Логическое НЕ
			print(f"a and b or c: {a and b or c}") # Комбинация операторов
			# Приоритет операторов
			result1 = not a and b or c
			result2 = not (a and b) or c
			print(f"Разные приоритеты: {result1} vs {result2}")

Тема 3 Организация повторений	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1. Знать: Методы и средства проектирования программных интерфейсов ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	1. Преобразование температуры с использованием цикла for Напишите код, который бы переводил значения температуры, взятые по шкале Цельсия в соответствующие значения шкалы Фаренгейт, используя цикл for по элементам списка. Формула перевода: F = (9/5) × C + 32 python celsius_temps = [0, 10, 20, 30, 40, 100] fahrenheit_temps = [] for celsius in celsius_temps: fahrenheit = (9/5) * celsius + 32 fahrenheit_temps.append(fahrenheit) print(f"Температуры по Цельсию: {celsius_temps}") print(f"Температуры по Фаренгейту: {fahrenheit_temps}") 2. Вычисление значений функции (компактный синтаксис) Значения переменной x заданы в виде списка: x = [1, 2, 3, 4, 5].

Тема	Код компетенции	Индикаторы достижения компетенции	Типовое задание и решение

			Используя компактный синтаксис Python, напишите код (в одну строку) который бы выводил в виде списка значения функции y = x².
			python
			x = [1, 2, 3, 4, 5]
			y = [value**2 for value in x]
			print(f"x = {x}")
			print(f"y = x² = {y}")
			3. Вычисление среднего арифметического
			Напишите код программы, которая будет запрашивать у пользователя значения и вычислять по ним среднее арифметическое.
			python
			values = []
			print("Введите числа для вычисления среднего (для завершения введите 'stop'):")
			while True:
			user_input = input("Введите число: ")
			if user_input.lower() == 'stop':
			break
			try:
			number = float(user_input)
			values.append(number)
			except ValueError:
			print("Ошибка! Введите корректное число или 'stop'")
			if values:
			average = sum(values) / len(values)
			print(f"Введенные значения: {values}")
			print(f"Среднее арифметическое: {average:.2f}")
			else:
			print("Не введено ни одного числа")

Тема 3 Организация	ПК-2. Способен осуществлять оценку	ИПК-2.1 Знать: Возможности средств	4. Вычисление значения функции по запросу пользователя Напишите код программы, которая будет запрашивать у пользователя

Тема	Код компетенции	Индикаторы достижения компетенции	Типовое задание и решение

повторений (продолжение)	времени и трудоёмкости	разработки	функцию f(x), значение переменной x и выводить значение функции f(x).
			python
			print("Доступные функции:")
			print("1. x**2 (квадрат)")
			print("2. x**3 (куб)")
			print("3. 2*x + 1 (линейная)")
			print("4. x**2 + 2*x + 1 (квадратичная)")
			
			function_choice = input("\nВыберите номер функции (1-4): ")
			x = float(input("Введите значение x: "))
			
			if function_choice == '1':
			 result = x**2
			 formula = "x²"
			elif function_choice == '2':
			 result = x**3
			 formula = "x³"
			elif function_choice == '3':
			 result = 2*x + 1
			 formula = "2x + 1"
			elif function_choice == '4':
			 result = x**2 + 2*x + 1
			 formula = "x² + 2x + 1"
			else:
			 print("Неверный выбор функции")
			 exit()
			print(f"\nФункция: f(x) = {formula}")
			print(f"При x = {x}")
			print(f"Результат: f({x}) = {result}")
			5. Создание вложенного списка значений функции
			Значения переменной x заданы в виде списка x = [1, 2, 3, 4, 5].
			Необходимо получить соответствующий список значений функции y = x², а результат вывести на экран в виде вложенного списка вида: [[x_1, y_1],

<table>
<tr><th>Тема</th><th>Код компетенции</th><th>Индикаторы достижения компетенции</th><th>Типовое задание и решение</th></tr>
<tr><td></td><td></td><td></td><td>[x₂, y₂], ...].
```python
x = [1, 2, 3, 4, 5]
result = [[xi, xi**2] for xi in x]
print("Вложенный список [x, y]:")
print(result)
print("\nТабличное представление:")
print(" x | y=x²")
print("------|-------")
for pair in result:
  print(f" {pair[0]:2} | {pair[1]:4}")
Вложенный список [x, y]:
[[1, 1], [2, 4], [3, 9], [4, 16], [5, 25]]
Табличное представление:
  x | y=x²
-----|-------
  1 | 1
  2 | 4
  3 | 9
  4 | 16
  5 | 25</td></tr>
<tr><td>Тема 4
Упорядоченные типы данных</td><td>ПК-6. Способен осуществлять проектирование БД</td><td>ИПК-6.1 Знать: Методы проектирования БД
ИПК-6.2 Уметь: Применять методы проектирования</td><td>1. Сравнительный анализ методов работы со списками и строками
Заполните таблицу сравнения операций со списками и строками:
<table><tr><th>Операция</th><th>Списки</th><th>Строки</th><th>Изменяемость</th></tr><tr><td>Добавление</td><td>list.append(x)</td><td>str + str</td><td>Только списки</td></tr><tr><td>Изменение</td><td>list[i] = value</td><td>Не поддерживается</td><td>Только списки</td></tr><tr><td>Срезы</td><td>list[start:stop]</td><td>str[start:stop]</td><td>Оба типа</td></tr><tr><td>Длина</td><td>len(list)</td><td>len(str)</td><td>Оба типа</td></tr><tr><td>Поиск</td><td>element in list</td><td>substring in str</td><td>Оба типа</td></tr></table>
2. Исследование методов работы с кортежами</td></tr>
</table>

Тема	Код компетенции	Индикаторы достижения компетенции	Типовое задание и решение

			Разработайте схемы использования кортежей и их сравнение со списками:
			python
			# Исследование кортежей
			tuple_example = (1, 2, 3, "hello")
			list_example = [1, 2, 3, "hello"]
			
			print(f"Кортеж: {tuple_example}, тип: {type(tuple_example)}")
			print(f"Список: {list_example}, тип: {type(list_example)}")
			
			# Сравнение производительности
			import sys
			print(f"Размер кортежа в памяти: {sys.getsizeof(tuple_example)} байт")
			print(f"Размер списка в памяти: {sys.getsizeof(list_example)} байт")
			
			# Распаковка кортежа
			a, b, c, d = tuple_example
			print(f"Распаковка: a={a}, b={b}, c={c}, d={d}")
			
			3. Анализ эффективности строковых операций
			Проведите исследование времени выполнения различных строковых операций:
			python
			import time
			
			def measure_string_operations():
			# Конкатенация vs join
			start = time.time()
			result = ""
			for i in range(1000):
			result += str(i)
			concat_time = time.time() - start

Тема	Код компетенции	Индикаторы достижения компетенции	Типовое задание и решение
			start = time.time() result = "".join(str(i) for i in range(1000)) join_time = time.time() - start return concat_time, join_time concat, join = measure_string_operations() print(f"Время конкатенации: {concat:.4f} сек") print(f"Время join(): {join:.4f} сек") Результаты эксперимента: • Конкатенация: ~0.0008 сек • join(): ~0.0002 сек

Тема 5
Неупорядоченные типы данных
	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1. Знать:
		Методы и средства проектирования программных интерфейсов
		ИПК-7.2. Уметь:
		Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов
			1. Проектирование структур данных для предметных областей
			Разработайте структуры данных для следующих предметных областей:
			python
			# Система управления студентами
			student_system = {
			"students": {
			"001": {"name": "Иван", "age": 20, "grades": {"math": 5, "physics": 4}},
			"002": {"name": "Мария", "age": 19, "grades": {"math": 4, "physics": 5}}
			},
			"subjects": {"math", "physics", "programming"},
			"statistics": {}
			}
			
			# Интернет-магазин
			online_store = {
			"products": {
			"123": {"name": "Ноутбук", "price": 50000, "category": "электроника"},
			"456": {"name": "Книга", "price": 500, "category": "литература"}

Тема	Код компетенции	Индикаторы достижения компетенции	Типовое задание и решение

			},
			"categories": {"электроника", "литература", "одежда"},
			"orders": set()
			}
			2. Сравнительный анализ множеств и словарей
			Заполните таблицу сравнения множеств и словарей:
			• Структура: Множество - неупорядоченные уникальные элементы; Словарь - пары ключ-значение
			• Изменяемость: Оба - изменяемые структуры
			• Доступ: Множество - только проверка наличия; Словарь - по ключу
			• Основные операции: Множество - add(), remove(), union(); Словарь - get(), items(), keys()
			• Применение: Множество - уникальные элементы, операции теории множеств; Словарь - ассоциативные массивы, JSON-like структуры
			3. Исследование операций с множествами
			Изучите и примените операции теории множеств:
			python
			set_a = {1, 2, 3, 4, 5}
			set_b = {4, 5, 6, 7, 8}
			
			print(f"Множество A: {set_a}")
			print(f"Множество B: {set_b}")
			print(f"Объединение: {set_a | set_b}")
			print(f"Пересечение: {set_a & set_b}")
			print(f"Разность A-B: {set_a - set_b}")
			print(f"Разность B-A: {set_b - set_a}")
			print(f"Симметричная разность: {set_a ^ set_b}")
			
			# Практическое применение - удаление дубликатов
			duplicates = [1, 2, 2, 3, 4, 4, 5]
			unique = list(set(duplicates))
			print(f"Исходный список: {duplicates}")
			print(f"Уникальные элементы: {unique}")

Тема	Код компетенции	Индикаторы достижения компетенции	Типовое задание и решение

Критерии оценивания самостоятельной работы:

Оценка	Критерии оценивания
отлично	выставляется если работа носит научно-исследовательский характер, проанализирован и сделан сравнительный анализ нескольких литературных источников, приведены примеры
хорошо	выставляется если проанализирован и сделан сравнительный анализ нескольких литературных источников, приведены примеры
удовлетворительно	выставляется если проведен сравнительный анализ научно-методической литературы, приведены примеры
неудовлетворительно	выставляется если работа прошла проверку на антиплагиат и соответствует требованиям оформления

4.5. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ ДЛЯ КУРСОВЫХ РАБОТ

Курсовая работа позволяет выявить степень владения базовыми знаниями, умениями и навыками, необходимыми для обучения, и определить уровень владения новым материалом.

Примерные индивидуальные задания (темы) для курсовых работ:

КУРСОВАЯ РАБОТА НЕ ПРЕДУСМОТРЕНА УЧЕБНЫМ ПЛАНОМ

Критерии оценивания курсовой работы:

Оценка	Критерии оценивания
отлично	Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Представлены результаты структурированного и логически последовательного обзора литературных и иных источников по теме исследования. Структура курсовой работы логически и методически выдержана. Верно определены исходные данные для расчетов. Все аналитические расчеты выполнены верно, корректно применены методы экономического анализа, не нарушена методика анализа предмета исследования. Все выводы и предложения убедительно аргументированы. При защите курсовой работы обучающийся правильно и уверенно отвечает на вопросы преподавателя, демонстрирует глубокое знание теоретического материала, способен аргументировать собственные утверждения и выводы
хорошо	Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Представлены результаты структурированного и логически последовательного обзора литературных и иных источников по теме исследования. Структура курсовой работы логически и методически выдержана. Верно определены исходные данные для расчетов. В расчетах допускаются незначительные (не искажающие общего итога оценки) погрешности/ошибки. Большинство выводов и предложений аргументировано, корректно применены методы экономического анализа, не нарушена методика анализа предмета исследования. Оформление курсовой работы и полученные результаты в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две несущественные ошибки в использовании терминов, в построенных диаграммах и схемах, в оформлении таблиц. Наличествует незначительное количество грамматических и/или стилистических ошибок. При защите курсовой работы обучающийся правильно и уверенно отвечает на большинство вопросов преподавателя, демонстрирует хорошее знание теоретического материала, но не всегда способен аргументировать собственные утверждения и выводы. При наводящих вопросах преподавателя исправляет ошибки в ответе

удовлетворительно	Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Результаты обзора литературных и иных источников представлены недостаточно полно, недостаточно логично и последовательно. Верно определены исходные данные для расчетов, но имеются грубые ошибки в расчетах. Аргументация выводов и предложений слабая или отсутствует. Экономические выводы носят констатирующий (описательный) характер. Имеются одно-два существенных отклонений от требований в оформлении курсовой работы. Полученные результаты в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две существенных ошибки в использовании терминов, в построенных диаграммах и схемах. Много грамматических и/или стилистических ошибок. При защите курсовой работы обучающийся допускает грубые ошибки при ответах на вопросы преподавателя, демонстрирует слабое знание теоретического материала, в большинстве случаев не способен уверенно аргументировать собственные утверждения и выводы
неудовлетворительно	Содержание курсовой работы не соответствует заданию, содержащемуся в методических указаниях, и плану. Неверно определены исходные данные для расчетов, неверно и не корректно применены методы экономического анализа. Экономические выводы содержат неверную экономическую оценку. Имеются более двух существенных отклонений от требований в оформлении курсовой работы. Большое количество существенных ошибок по сути работы, много грамматических и стилистических ошибок и др. Полученные результаты не отвечают требованиям, изложенным в методических указаниях. При защите курсовой работы обучающийся демонстрирует слабое понимание программного материала, студент не может защитить свои решения, допускает грубые фактические ошибки при ответах на поставленные вопросы или вовсе не отвечает на них. Курсовая работа не представлена преподавателю. Обучающийся не явился на защиту курсовой работы

5. МАТЕРИАЛЫ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Экзаменационные вопросы по дисциплине "Язык программирования Python"
Семестр 5 (Зачет с оценкой)

1. Что включает в себя экосистема Python и каковы этапы разработки программы?
2. Какие инструменты разработки используются для создания программ на Python и как выглядит жизненный цикл программы?
3. Какова структура программы на Python и для чего используются переменные и типы данных?
4. Какие типы данных существуют в языке Python и как объявляются переменные?
5. Какие операторы поддерживаются в языке Python и как определяется их приоритет?
6. Какие простые типы данных существуют в Python? Охарактеризуйте операции ввода и вывода информации.
7. Как организованы комментарии в Python и какие виды комментариев существуют?
8. Что такое логические переменные в Python и какие значения они могут принимать?
9. Какие операторы отношения существуют в Python и как они используются в логических выражениях?
10. Опишите все варианты инструкции if в Python (if, if-else, if-elif-else).
11. Как работают вложенные условные выражения в Python и в каких случаях их целесообразно использовать?
12. Какова структура и синтаксис циклической инструкции while в Python?
13. Какова структура и синтаксис циклической инструкции for в Python?

14. Как используются логические переменные в условии цикла? Приведите примеры.
15. Что такое строки в Python? Какие методы и функции для работы со строками существуют?
16. Назовите принципы индексации и создания срезов в упорядоченных типах данных Python.
17. Что такое списки в Python? Какие методы и функции для работы со списками существуют?
18. Что такое кортежи в Python и чем они отличаются от списков?
19. Как использовать цикл for для редактирования строк в Python?
20. Как осуществляется преобразование кортежей в другие типы данных и обратно?

Семестр 6 (Экзамен)

1. Что такое множества в Python и какие операции с ними можно выполнять?
2. Опишите основные операции с множествами: объединение, пересечение, разность, симметрическая разность.
3. Что такое словари в Python и как они организованы?
4. Какие основные операции со словарями существуют в Python (добавление, удаление, изменение элементов)?
5. Как использовать множества в логическом контексте? Приведите примеры.
6. Что такое словари со смешанными значениями и как с ними работать?
7. Приведите примеры вложенных структур данных в Python (списки списков, словари словарей и т.д.).
8. Как осуществляется обращение к элементам вложенных структур данных?
9. Какие способы организации вывода вложенных структур данных существуют в Python?
10. Как организовано модульное программирование и импорт в Python?
11. Как определяется функция в Python? Опишите синтаксис определения функции.
12. Как вызывается функция в Python и как передаются аргументы?
13. Что такое параметры функции и какие виды параметров существуют (позиционные, именованные, со значениями по умолчанию)?
14. Что такое области видимости в Python и как работает пространство имен?
15. Что включает в себя стандартная библиотека Python и как использовать её модули?
16. Как осуществляется работа с файлами в языке Python? Опишите основные операции.
17. Какие режимы открытия файлов существуют в Python и для чего они используются?
18. Что такое символы конца строки и как они обрабатываются при работе с файлами?
19. Как обрабатываются исключения в Python и какие типы исключений существуют?
20. Каковы основные принципы проектирования структур данных в Python? Как оценивается сложность операций с различными структурами данных?

Критерии оценивания зачета с оценкой/экзамена:

Оценка	Критерии оценивания
зачтено	Обучающийся должен: уметь строить ответ в соответствии со структурой излагаемого вопроса; продемонстрировать прочное, достаточно полное усвоение знаний программного материала; продемонстрировать знание основных теоретических понятий; правильно формулировать определения; последовательно, грамотно и логически стройно изложить теоретический материал;

	продемонстрировать умения самостоятельной работы с литературой; уметь сделать достаточно обоснованные выводы по излагаемому материалу.
не зачтено	Обучающийся демонстрирует: незнание значительной части программного материала; не владение понятийным аппаратом дисциплины; существенные ошибки при изложении учебного материала; неумение строить ответ в соответствии со структурой излагаемого вопроса; неумение делать выводы по излагаемому материалу.

Критерии оценивания зачета с оценкой/экзамена:

Оценка	Критерии оценивания
отлично	Выставляется, если обучающимся правильно и полностью раскрыто содержание материала в пределах программы, чётко и правильно даны определения и раскрыто содержание понятий, точно использованы научные и технические термины, в ответе использованы ранее приобретённые теоретические знания, сделаны необходимые выводы и обобщения
хорошо	Выставляется, если обучающимся раскрыто основное содержание материала в пределах программы, даны определения и раскрыто содержание понятий, в ответе использованы ранее приобретённые теоретические знания, сделаны необходимые выводы и обобщения, но присутствуют незначительные нарушения в последовательности изложения, имеются одна-две неточности в содержании ответа
удовлетворительно	Выставляется, если обучающимся содержание учебного материала изложено фрагментарно, не всегда последовательно, не даны определения, не раскрыто содержание понятий, или они изложены с ошибками, допускаются ошибки и неточности в использовании научной терминологии, отсутствуют выводы и обобщения из предыдущего материала, или возможны ошибки в их изложении
неудовлетворительно	Выставляется, если обучающимся основное содержание учебного материала не раскрыто, не даются ответы на основные вопросы, допускаются грубые ошибки в определении понятий, в использовании терминологии, отсутствуют выводы и обобщения

6. МАТЕРИАЛЫ ДЛЯ ДИАГНОСТИЧЕСКОЙ РАБОТЫ

Задания для диагностической работы должны обеспечивать оценку полностью или частично сформированных компетенций. Каждое задание должно быть привязано к тому или иному индикатору сформированности компетенций.

При формировании заданий для диагностической работы необходимо использовать тестовые задания следующих типов:

Тип задания 1. Задания закрытого типа на установление соответствия.

Тип задания 2. Задания закрытого типа на установление последовательности.

Тип задания 3. Задания комбинированного типа, предполагающие выбор одного правильного ответа из предложенных

Тип задания 4. Задания комбинированного типа, предполагающие выбор нескольких ответов из предложенных

Тип задания 5. Задания открытого типа с развернутым ответом.

Все типы заданий должны быть представлены не менее одного раза.

<table>
<tr><th rowspan="2">№ п/п</th><th rowspan="2">Тема занятия</th><th rowspan="2">Код компетенции</th><th rowspan="2">Индикатор</th><th rowspan="2">Тип задания</th><th colspan="2">Задание</th><th rowspan="2">Уровень освоения компетенции</th></tr>
<tr><th>Вариант 1</th><th>Вариант 2</th></tr>
<tr><td>1.</td><td>Тема 1 Возможности языка программирования Python</td><td>П К - 2 Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению</td><td>ИПК-2.1 Знать: Возможности современных и перспективных средств разработки программных продуктов и технических средств</td><td>1. Задания закрытого типа на установление соответствия</td><td>1. Соотнесите типы данных в Python с их описанием:
<table><tr><th>Буква</th><th>Тип данных</th><th>Номер</th><th>Описание</th></tr><tr><td>А</td><td>int</td><td>1</td><td>Неизменяемая последовательность символов</td></tr><tr><td>Б</td><td>float</td><td>2</td><td>Изменяемая упорядоченная коллекция</td></tr><tr><td>В</td><td>str</td><td>3</td><td>Целочисленный тип</td></tr><tr><td>Г</td><td>list</td><td>4</td><td>Вещественный тип</td></tr></table>Запишите выбранные цифры:
<table><tr><td>А</td><td>Б</td><td>В</td><td>Г</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>Решение:
<table><tr><td>А</td><td>Б</td><td>В</td><td>Г</td></tr><tr><td>3</td><td>4</td><td>1</td><td>2</td></tr></table>Ответ: А-3, Б-4, В-1, Г-2</td><td>1. Соотнесите типы данных с их характеристиками:
<table><tr><th>Буква</th><th>Тип данных</th><th>Номер</th><th>Характеристика</th></tr><tr><td>А</td><td>tuple</td><td>1</td><td>Изменяемый словарь</td></tr><tr><td>Б</td><td>dict</td><td>2</td><td>Неизменяемая последовательность</td></tr><tr><td>В</td><td>set</td><td>3</td><td>Коллекция уникальных элементов</td></tr><tr><td>Г</td><td>bool</td><td>4</td><td>Логический тип</td></tr></table>Запишите выбранные цифры:
<table><tr><td>А</td><td>Б</td><td>В</td><td>Г</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>Решение:
<table><tr><td>А</td><td>Б</td><td>В</td><td>Г</td></tr><tr><td>2</td><td>1</td><td>3</td><td>4</td></tr></table>Ответ: А-2, Б-1, В-3, Г-4</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>2</td><td></td><td></td><td>И П К - 2 . 2 Уметь: Выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению</td><td>3. Задания комбинированного типа с выбором одного ответа</td><td>2. Какой тип данных следует использовать для хранения возраста человека?
1. float
2. str
3. int
4. bool
Обоснование:
Ответ: 3) int - так как возраст представляет целое число</td><td>2. Какой тип данных наиболее эффективен для хранения математических констант?
1. int
2. float
3. str
4. bool
Обоснование:
Ответ: 3) char - наиболее эффективен для хранения true/false</td><td>Повышенный (3-5 минут)</td></tr>
</table>

3		П К - 7 Способен осуществлять проектирование программных интерфейсов	И П К - 7 . 1 . Знать: Методы и средства проектирования программных интерфейсов	5. Задания открытого типа с развернутым ответом	3. Напишите программу, которая определяет тип данных переменной: Решение: python x = 3.14 print(f"Тип переменной: {type(x)}")	3. Напишите программу, которая демонстрирует преобразование типов: Решение: python num_str = "123" num_int = int(num_str) print(f"Результат: {num_int}, тип: {type(num_int)}")	
4	Тема 2 Организация ветвлений	**ПК-6** Способен осуществ-лять проектирование баз данных	ИПК-6.1 Знать: Методы и средства проектирования баз данных	2. Задания закрытого типа на установление последовательности	4. Расположите этапы выполнения программы в правильной последовательности: 1. Интерпретация кода 2. Написание кода 3. Выполнение инструкций 4. Сохранение файла Запишите последовательность: Ответ: 2-4-1-3	4. Расположите операторы в порядке выполнения: 1. print("Результат:", x) 2. x = 10 3. x = x * 2 4. import math Запишите последовательность Ответ: 4-2-3-1	**Базовый (1-3 минуты)**
5			ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	4. Задания комбинированного типа с выбором нескольких ответов	5. Какие из перечисленных конструкций являются базовыми в Python? 1. Ветвление (if-elif-else) 2. Наследование классов 3. Циклы (for, while) 4. Импорт модулей 5. Декораторы Обоснование: Ответ: 1, 3, 4 - базовые конструкции языка Python	5. Какие операторы используются для организации циклов в Python? 1. if 2. for 3. switch 4. while 5. def Обоснование: Ответ: 2, 4 - операторы циклов в Python	**Базовый (1-3 минуты)**
6		П К - 7 Способен осуществлять проектирование программн	ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного	5. Задания открытого типа с развернутым ответом	6. Напишите программу, которая находит сумму всех четных чисел от 1 до 100: Ответ: python total = sum(range(2, 101, 2)) print(f"Сумма четных чисел: {total}")	6. Напишите программу, которая проверяет, является ли введенное число палиндромом. Ответ: python num = 12321 print("Палиндром" if str(num) == str(num)[::-1] else "Не палиндром")	**Повышенный (3-5 минут)**

<table>
<tr><td></td><td></td><td>ых интерфейсов</td><td>программного обеспечения, структур данных, баз данных, программных интерфейсов</td><td></td><td></td><td></td><td></td></tr>
<tr><td>7</td><td>Тема 4 Упорядоченные типы данных</td><td>П К - 2 Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению</td><td>ИПК-2.1 Знать: Возможности современных и перспективных средств разработки программных продуктов и технических средств</td><td>1. Задания закрытого типа на установление соответствия</td><td>7. Соотнесите операции со списками с их описанием:
<table><tr><td>А</td><td>append()</td><td>1</td><td>Добавление элемента в конец</td></tr><tr><td>Б</td><td>insert()</td><td>2</td><td>Вставка элемента по индексу</td></tr><tr><td>В</td><td>remove()</td><td>3</td><td>Удаление элемента по значению</td></tr><tr><td>Г</td><td>pop()</td><td>4</td><td>Удаление элемента по индексу</td></tr></table>Запишите выбранные цифры:
<table><tr><td>А</td><td>Б</td><td>В</td><td>Г</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>
Ответ: А-1, Б-2, В-3, Г-4</td><td>7. Соотнесите методы списков с их результатом:
<table><tr><td>Буква</td><td>Метод</td><td>Номер</td><td>Результат</td></tr><tr><td>А</td><td>sort()</td><td>1</td><td>Новый отсортированный список</td></tr><tr><td>Б</td><td>sorted()</td><td>2</td><td>Сортировка исходного списка</td></tr><tr><td>В</td><td>reverse()</td><td>3</td><td>Разворот списка</td></tr></table>Запишите выбранные цифры:
<table><tr><td>А</td><td>Б</td><td>В</td><td>Г</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>
Ответ: А-2, Б-1, В-3, Г-4</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>8</td><td></td><td></td><td>ИПК-2.2 Уметь: Выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению</td><td>3. Задания комбинированного типа с выбором одного ответа</td><td>8. Какой индекс имеет первый элемент списка в Python?
1. 0
2. 1
3. -1
4. first
Обоснование:
Ответ:1) 0 - индексация начинается с нуля</td><td>8. Что произойдет при обращении к несуществующему индексу списка?
1. Вернется None
2. Создастся новый элемент
3. Возникнет IndexError
4. Вернется 0
Обоснование:
Ответ:3) IndexError - выход за границы списка</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>9</td><td></td><td>П К - 7 Способен осуществлять проектирование программных интерфейсов</td><td>И П К - 7 . 1 . Знать: Методы и средства проектирования программных интерфейсов</td><td>5. Задания открытого типа с развернутым ответом</td><td>9. Напишите программу, которая находит среднее арифметическое элементов списка:
Ответ:
python
numbers = [1, 2, 3, 4, 5]
average = sum(numbers) / len(numbers)
print(f"Среднее: {average}")</td><td>9. Напишите программу, которая удаляет все отрицательные элементы из списка:
Ответ:
python
numbers = [1, -2, 3, -4, 5]
positive = [x for x in numbers if x >= 0]
print(f'Положительные: {positive}")</td><td>Повышенный (3-5 минут)</td></tr>
</table>

10	Тема 5 Неупорядоченные типы данных	П К - 6 Способен осуществлять проектирование баз данных	ИПК-6.1 Знать: Методы и средства проектирования баз данных	2. Задания закрытого типа на установление последовательности	10. Расположите этапы работы со словарем в правильной последовательности: 1. Создание словаря 2. Добавление элементов 3. Получение значений 4. Обновление данных Запишите последовательность: | | | | | Ответ: 1-2-3-4	10. Расположите операции с множеством в порядке выполнения: 1. Создание множества 2. Добавление элементов 3. Проверка наличия элемента 4. Удаление элемента Запишите последовательность: | | | | | Ответ: 1-2-3-4	**Базовый (1-3 минуты)**
11			ИПК-6.2 Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	4. Задания комбинированного типа с выбором нескольких ответов	11. Какие из перечисленных методов используются для работы со словарями? 1. keys() 2. append() 3. values() 4. add() 5. items() Обоснование: Ответ:1, 3, 5 - методы словарей в Python	11. Какие операции поддерживают множества? 1. Объединение 2. Пересечение 3. Добавление 4. Сортировка 5. Разность Обоснование: Ответ:1, 2, 3, 5 - основные операции с множествами данных	**Базовый (1-3 минуты)**

12		П К - 7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2. Уметь: Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	5. Задания открытого типа с развернутым ответом	12. Создайте словарь для хранения информации о студенте:
					Ответ:
					python
					student = {
					"name": "Иван",
					"age": 20,
					"grades": [5, 4, 3]
					}
					print(f"Студент: {student['name']}, средний балл:
					{sum(student['grades'])/len(student['grades'])}")
						12. Напишите программу для работы с множествами:
						Ответ:
						python
						set1 = {1, 2, 3, 4, 5}
						set2 = {4, 5, 6, 7, 8}
						print(f"Объединение: {set1 | set2}")
						print(f"Пересечение: {set1 & set2}")
							Высокий (5-10 минут)

<table>
<tr>
<td>13</td>
<td>Тема 7 Функции и модули</td>
<td>П К - 2 Способен осуществлять оценку времени и трудоёмкости и реализации требований к компьютерному программному обеспечению</td>
<td>ИПК-2.1 Знать: Возможности современных и перспективных средств разработки программных продуктов и технических средств</td>
<td>1. Задания закрытого типа на установление соответствия</td>
<td>13. Соотнесите понятия функций с их определениями:
<table>
<tr><td>А</td><td>Параметр</td><td>1</td><td>Переменная в определении функции</td></tr>
<tr><td>Б</td><td>Аргумент</td><td>2</td><td>Значение, передаваемое в функцию</td></tr>
<tr><td>В</td><td>Возврат</td><td>3</td><td>Результат работы функции</td></tr>
<tr><td>Г</td><td>Область видимости</td><td>4</td><td>Контекст доступности переменных</td></tr>
</table>
Запишите выбранные цифры:
<table>
<tr><td>А</td><td>Б</td><td>В</td><td>Г</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>
Ответ: А-1, Б-2, В-3, Г-4</td>
<td>13. Соотнесите элементы функции с их описанием:
<table>
<tr><td>А</td><td>def</td><td>1</td><td>Ключевое слово для определения функции</td></tr>
<tr><td>Б</td><td>return</td><td>2</td><td>Возврат результата</td></tr>
<tr><td>В</td><td>lambda</td><td>3</td><td>Анонимная функция</td></tr>
<tr><td>Г</td><td>import</td><td>4</td><td>Импорт модуля</td></tr>
</table>
Запишите выбранные цифры:
<table>
<tr><td>А</td><td>Б</td><td>В</td><td>Г</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>
Ответ: А-1, Б-2, В-3, Г-4</td>
<td>Базовый (1-3 минуты)</td>
</tr>
</table>

СЕМЕСТР 5

<table>
<tr><td rowspan="2">№ п/п</td><td rowspan="2">Тема занятия</td><td rowspan="2">Код компетенции</td><td rowspan="2">Индикатор</td><td rowspan="2">Тип задания</td><td colspan="2">Задание</td><td>Уровень освоения компетенции</td></tr>
<tr><td>Вариант 1</td><td>Вариант 2</td><td></td></tr>
<tr><td colspan="7">Тема 1 Возможности языка программирования Python</td><td></td></tr>
<tr>
<td>1</td>
<td>Тема 1.1</td>
<td>ПК-2</td>
<td>ИПК -2.1</td>
<td>1. Задания закрытого типа на установление соответствия</td>
<td>14. Задание: Соотнесите типы данных в Python с их описанием:
<table>
<tr><td>Пункт</td><td>Тип данных</td><td>Пункт</td><td>Описание</td></tr>
<tr><td>А</td><td>int</td><td>1</td><td>Неизменяемая последовательность символов</td></tr>
<tr><td>Б</td><td>float</td><td>2</td><td>Изменяемая упорядоченная коллекция</td></tr>
<tr><td>В</td><td>str</td><td>3</td><td>Целочисленный тип</td></tr>
</table></td>
<td>14. Задание: Соотнесите типы данных с их характеристиками:
<table>
<tr><td>Пункт</td><td>Тип данных</td><td>Пункт</td><td>Описание</td></tr>
<tr><td>А</td><td>tuple</td><td>1</td><td>Изменяемый словарь</td></tr>
<tr><td>Б</td><td>dict</td><td>2</td><td>Неизменяемая последовательность</td></tr>
<tr><td>В</td><td>set</td><td>3</td><td>Коллекция уникальных элементов</td></tr>
<tr><td>Г</td><td>bool</td><td>4</td><td>Логический тип</td></tr>
</table></td>
<td>Базовый (1-3 минуты)</td>
</tr>
</table>

<table>
<tr><th rowspan="2">№ п/п</th><th rowspan="2">Тема занятия</th><th rowspan="2">Код компетенции</th><th rowspan="2">Индикатор</th><th rowspan="2">Тип задания</th><th colspan="2">Задание</th><th>Уровень освоения компетенции</th></tr>
<tr><th>Вариант 1</th><th>Вариант 2</th><th></th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td><table><tr><td>Г</td><td>list</td><td>4</td><td>Вещественный тип</td></tr></table>
Ответ: А-3, Б-4, В-1, Г-2</td><td>Ответ: А-2, Б-1, В-3, Г-4</td><td></td></tr>
<tr><td>2</td><td>Тема 1</td><td>ПК-2</td><td>ИПК-2.2</td><td>3. Задания комбинированного типа с выбором одного ответа</td><td>15. Задание: Какой тип данных следует использовать для хранения возраста человека?
1. float 2. str 3. int 4. bool
Ответ: 3) int - так как возраст представляет целое число</td><td>15. Задание: Какой тип данных наиболее эффективен для хранения математических констант?
1. int 2. float 3. str 4. bool
Ответ: 2) float - для точного представления дробных чисел</td><td>Повышенный (3-5 минут)</td></tr>
<tr><td>3</td><td>Тема 1</td><td>ПК-7</td><td>ИПК-7.2</td><td>5. Задания открытого типа с развернутым ответом</td><td>16. Задание: Напишите программу, которая определяет тип данных переменной
Решение:
python
x = 3.14
print(f"Тип переменной: {type(x)}")</td><td>16. Задание: Напишите программу, которая демонстрирует преобразование типов
Решение:
python
num_str = "123"
num_int = int(num_str)
print(f"Результат: {num_int}, тип: {type(num_int)}")</td><td>Высокий (5-10 минут)</td></tr>
<tr><td>4</td><td>Тема 1</td><td>ПК-2</td><td>ИПК-2.1</td><td>1. Задания закрытого типа на установление</td><td>17. Задание: Соотнесите арифметические операции с их результатом для a=10, b=3:<table><tr><th>Пункт</th><th>Выражение</th><th>Пункт</th><th>Результат</th></tr><tr><td>А</td><td>a + b</td><td>1</td><td>13</td></tr><tr><td>Б</td><td>a - b</td><td>2</td><td>7</td></tr></table></td><td>17. Задание: Соотнесите операции с их описанием:<table><tr><th>Пункт</th><th>Оператор</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>//</td><td>1</td><td>Возведение в степень</td></tr><tr><td>Б</td><td>%</td><td>2</td><td>Целочисленн</td></tr></table></td><td>Базовый (1-3 минуты)</td></tr>
</table>

<table>
<tr><th>№ п/п</th><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th colspan="2">Задание</th><th>Уровень освоения компетенции</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Вариант 1</td><td>Вариант 2</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>соответствия</td><td><table><tr><td>В</td><td>a * b</td><td>3</td><td>30</td></tr><tr><td>Г</td><td>a % b</td><td>4</td><td>1</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>
Ответ: А-1, Б-4, В-2, Г-3</td><td><table><tr><td></td><td></td><td></td><td>ое деление</td></tr><tr><td>В</td><td>**</td><td>3</td><td>Остаток от деления</td></tr><tr><td>Г</td><td>/</td><td>4</td><td>Обычное деление</td></tr></table>
Ответ: А-2, Б-3, В-1, Г-4</td><td></td></tr>
<tr><td>5</td><td>Тема 1</td><td>ПК-2</td><td>ИПК -2.2</td><td>3. Задания комбинированного типа с выбором одного ответа</td><td>18. Задание: Какая операция имеет наивысший приоритет в выражении 2 + 3 * 4 ** 2?
1. + 2. * 3.
Ответ:** 3) ** - возведение в степень имеет наивысший приоритет</td><td>18. Задание: Что выведет код: print(2 * 3 + 4 ** 2)?
1. 22 2. 100 3. 28
Ответ: 1) 22 - сначала 4**2=16, затем 2*3=6, потом 6+16=22</td><td>Повышенный (3-5 минут)</td></tr>
<tr><td colspan="7">Тема 2 Организация ветвлений</td><td></td></tr>
<tr><td>6</td><td>Тема 2</td><td>ПК-6</td><td>ИПК -6.1</td><td>2. Задания закрытого типа на установление последовательности</td><td>19. Задание: Расположите этапы выполнения программы в правильной последовательности:
1. Интерпретация кода
2. Написание кода
3. Выполнение инструкций
4. Сохранение файла
Ответ: 2-4-1-3</td><td>19. Задание: Расположите операторы в порядке выполнения:
1. print("Результат:", x)
2. x = 10
3. x = x * 2
4. import math
Ответ: 4-2-3-1</td><td>Повышенный (3-5 минут)</td></tr>
<tr><td>7</td><td>Тема</td><td>ПК-6</td><td>ИПК</td><td>4.</td><td>20. Задание: Какие из перечисленных</td><td>20. Задание: Какие операторы используются для</td><td>Базовый (1-3</td></tr>
</table>

<table>
<tr><th rowspan="2">№ п/п</th><th rowspan="2">Тема занятия</th><th rowspan="2">Код компетенции</th><th rowspan="2">Индикатор</th><th rowspan="2">Тип задания</th><th colspan="2">Задание</th><th>Уровень освоения компетенции</th></tr>
<tr><th>Вариант 1</th><th>Вариант 2</th><th></th></tr>
<tr><td></td><td>2</td><td></td><td>-6.2</td><td>Задания комбинированного типа с выбором нескольких ответов</td><td>конструкций являются базовыми в Python?
1. Ветвление (if-elif-else)
2. Наследование классов
3. Циклы (for, while)
4. Импорт модулей
5. Декораторы
Ответ: 1, 3, 4 - базовые конструкции языка Python</td><td>организации циклов в Python?
1. if
2. for
3. switch
4. while
5. def
Ответ: 2, 4 - операторы циклов в Python</td><td>минуты)</td></tr>
<tr><td>8</td><td>Тема 2</td><td>ПК-7</td><td>ИПК-7.2</td><td>5. Задания открытого типа с развернутым ответом</td><td>21. Задание: Напишите программу, которая находит сумму всех четных чисел от 1 до 100
Решение:
python
total = sum(range(2, 101, 2))
print(f"Сумма четных чисел: {total}")</td><td>21. Задание: Напишите программу, которая проверяет, является ли введенное число палиндромом
Решение:
python
num = 12321
print("Палиндром" if str(num) == str(num)[::-1] else "Не палиндром")</td><td>Высокий (5-10 минут)</td></tr>
<tr><td>9</td><td>Тема 2</td><td>ПК-6</td><td>ИПК-6.1</td><td>1. Задания закрытого типа на установление соответствия</td><td>22. Задание: Соотнесите операторы сравнения с их значениями:
<table><tr><th>Пункт</th><th>Оператор</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>==</td><td>1</td><td>Не равно</td></tr><tr><td>Б</td><td>!=</td><td>2</td><td>Равно</td></tr><tr><td>В</td><td>>=</td><td>3</td><td>Меньше или равно</td></tr><tr><td>Г</td><td><=</td><td>4</td><td>Больше или равно</td></tr></table>Ответ: А-2, Б-1, В-4, Г-3</td><td>22. Задание: Соотнесите логические операторы с их значениями:
<table><tr><th>Пункт</th><th>Оператор</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>and</td><td>1</td><td>Логическое ИЛИ</td></tr><tr><td>Б</td><td>or</td><td>2</td><td>Логическое И</td></tr><tr><td>В</td><td>not</td><td>3</td><td>Логическое НЕ</td></tr></table>Ответ: А-2, Б-1, В-3</td><td>Базовый (1-3 минуты)</td></tr>
</table>

№ п/п	Тема занятия	Код компетенции	Индикатор	Тип задания	Задание		Уровень освоения компетенции
					Вариант 1	Вариант 2	
10	Тема 2	ПК-7	ИПК-7.2	3. Задания комбинированного типа с выбором одного ответа	23. Задание: Что выведет код: print(5 > 3 and 2 < 4) 1. True 2. False 3. Ошибка Ответ: 1) True - оба условия истинны	23. Задание: Что выведет код: print(not (5 < 3) or (2 > 4)) 1. True 2. False 3. Ошибка Ответ: 1) True - not(False) = True, True or False = True	**Повышенный (3-5 минут)**
Тема 3 Организация повторений							
11	Тема 3	ПК-7	ИПК-7.1	2. Задания закрытого типа на установление последовательности	24. Задание: Расположите этапы выполнения цикла for: A) Выполнение тела цикла B) Переход к следующему элементу C) Получение первого элемента D) Проверка наличия элементов Ответ: C-D-A-B-D	24. Задание: Расположите этапы выполнения цикла while: A) Проверка условия B) Выполнение тела цикла C) Инициализация переменной D) Изменение переменной Ответ: C-A-B-D-A	**Базовый (1-3 минуты)**
12	Тема 3	ПК-6	ИПК-6.1	4. Задания комбинированного типа с выбором нескольких ответов	25. Задание: Какие операторы управления циклом существуют в Python? 1. break 2. continue 3. pass 4. return 5. exit Ответ: 1, 2, 3 - операторы управления	25. Задание: Какие типы циклов поддерживает Python? 1. for 2. while 3. do-while 4. foreach 5. repeat-until Ответ: 1, 2 - основные типы циклов в Python	**Повышенный (3-5 минут)**

<table>
<tr><th rowspan="2">№ п/п</th><th rowspan="2">Тема занятия</th><th rowspan="2">Код компетенции</th><th rowspan="2">Индикатор</th><th rowspan="2">Тип задания</th><th colspan="2">Задание</th><th rowspan="2">Уровень освоения компетенции</th></tr>
<tr><th>Вариант 1</th><th>Вариант 2</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>циклом</td><td></td><td></td></tr>
<tr><td>13</td><td>Тема 3</td><td>ПК-6</td><td>ИПК-6.2</td><td>5. Задания открытого типа с развернутым ответом</td><td>26. Задание: Напишите программу, которая выводит таблицу умножения для числа 5
Решение:
python
for i in range(1, 11):
print(f"5 × {i} = {5*i}")</td><td>26. Задание: Напишите программу, которая запрашивает число и выводит его таблицу умножения
Решение:
python
n = int(input("Введите число: "))
for i in range(1, 11):
print(f"{n} × {i} = {n*i}")</td><td>Высокий (5-10 минут)</td></tr>
<tr><td>14</td><td>Тема 3</td><td>ПК-7</td><td>ИПК-7.2</td><td>1. Задания закрытого типа на установление соответствия</td><td>27. Задание: Соотнесите циклы с их характеристиками:
<table><tr><th>Пункт</th><th>Ключевое слово</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>for</td><td>1</td><td>Выполняется пока условие истинно</td></tr><tr><td>Б</td><td>while</td><td>2</td><td>Итерируется по последовательности</td></tr><tr><td>В</td><td>break</td><td>3</td><td>Пропуск итерации</td></tr><tr><td>Г</td><td>continue</td><td>4</td><td>Выход из цикла</td></tr></table>Ответ: А-2, Б-1, В-4, Г-3</td><td>27. Задание: Соотнесите конструкции с их описанием:
<table><tr><th>Пункт</th><th>Выражение</th><th>Пункт</th><th>Результат</th></tr><tr><td>А</td><td>range(5)</td><td>1</td><td>Числа 0,1,2,3,4</td></tr><tr><td>Б</td><td>range(1,6)</td><td>2</td><td>Числа 1,2,3,4,5</td></tr><tr><td>В</td><td>range(0,10,2)</td><td>3</td><td>Числа 0,2,4,6,8</td></tr></table>Ответ: А-1, Б-2, В-3</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>15</td><td>Тема 3</td><td>ПК-6</td><td>ИПК-6.1</td><td>3. Задания комбинированного типа с</td><td>28. Задание: Сколько итераций выполнит цикл: for i in range(5):
1. 4 2. 5 3. 6
Ответ: 2) 5 - диапазон 0-4 включительно</td><td>28. Задание: Сколько итераций выполнит цикл: for i in range(2, 10, 2):
1. 4 2. 5 3. 6
Ответ: 1) 4 - числа 2,4,6,8</td><td>Повышенный (3-5 минут)</td></tr>
</table>

<table>
<tr><td rowspan="2">№ п/п</td><td rowspan="2">Тема занятия</td><td rowspan="2">Код компетенции</td><td rowspan="2">Индикатор</td><td rowspan="2">Тип задания</td><td colspan="2">Задание</td><td>Уровень освоения компетенции</td></tr>
<tr><td>Вариант 1</td><td>Вариант 2</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>выбором одного ответа</td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Тема 4 Упорядоченные типы данных</td><td></td></tr>
<tr><td>16</td><td>Тема 4</td><td>ПК-2</td><td>ИПК-2.1</td><td>1. Задания закрытого типа на установление соответствия</td><td>29. Задание: Соотнесите операции со списками с их описанием:
<table><tr><th>Пункт</th><th>Метод</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>append()</td><td>1</td><td>Добавление элемента в конец</td></tr><tr><td>Б</td><td>insert()</td><td>2</td><td>Вставка элемента по индексу</td></tr><tr><td>В</td><td>remove()</td><td>3</td><td>Удаление элемента по значению</td></tr><tr><td>Г</td><td>pop()</td><td>4</td><td>Удаление элемента по индексу</td></tr></table>Ответ: А-1, Б-2, В-3, Г-4</td><td>29. Задание: Соотнесите методы списков с их результатом:
<table><tr><th>Пункт</th><th>Метод</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>sort()</td><td>1</td><td>Новый отсортированный список</td></tr><tr><td>Б</td><td>sorted()</td><td>2</td><td>Сортировка исходного списка</td></tr><tr><td>В</td><td>reverse()</td><td>3</td><td>Разворот списка</td></tr></table>Ответ: А-2, Б-1, В-3</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>17</td><td>Тема 4</td><td>ПК-2</td><td>ИПК-2.2</td><td>3. Задания комбинированного типа с выбором одного ответа</td><td>30. Задание: Какой индекс имеет первый элемент списка в Python?
1. 0
2. 1
3. -1
4. first
Ответ: 1) 0 - индексация начинается с нуля</td><td>30. Задание: Что произойдет при обращении к несуществующему индексу списка?
1. Вернется None
2. Создастся новый элемент
3. Возникнет IndexError
4. Вернется 0
Ответ: 3) IndexError - выход за границы списка</td><td>Повышенный (3-5 минут)</td></tr>
<tr><td>18</td><td>Тема 4</td><td>ПК-7</td><td>ИПК-7.1</td><td>5. Задания открытог</td><td>31. Задание: Напишите программу, которая находит среднее арифметическое элементов списка</td><td>31. Задание: Напишите программу, которая удаляет все отрицательные элементы из списка
Решение:</td><td>Высокий (5-10 минут)</td></tr>
</table>

№ п/п	Тема занятия	Код компетенции	Индикатор	Тип задания	Задание		Уровень освоения компетенции
					Вариант 1	Вариант 2	
				о типа с развернутым ответом	Решение: python numbers = [1, 2, 3, 4, 5] average = sum(numbers) / len(numbers) print(f"Среднее: {average}")	python numbers = [1, -2, 3, -4, 5] positive = [x for x in numbers if x >= 0] print(f"Положительные: {positive}")	
19	Тема 4	ПК-2	ИПК-2.1	4. Задания комбинированного типа с выбором нескольких ответов	32. Задание: Какие методы изменяют исходный список? 1. append() 2. sorted() 3. insert() 4. remove() 5. len() Ответ: 1, 3, 4 - методы, изменяющие список	32. Задание: Какие операции поддерживают и строки, и списки? 1. Срезы 2. Конкатенация + 3. Изменение по индексу 4. Метод append() 5. Метод find() Ответ: 1, 2 - общие операции для строк и списков	**Базовый (1-3 минуты)**
20	Тема 4	ПК-2	ИПК-2.2	5. Задания открытого типа с развернутым ответом	33. Задание: Напишите программу для работы со строками: преобразование в верхний регистр и разделение на слова Решение: python text = "Hello World" print(text.upper()) print(text.split())	33. Задание: Напишите программу для работы со строками: удаление пробелов и замена подстроки Решение: python s = " Python " print(s.strip()) print(s.replace('Python', 'Java'))	**Высокий (5-10 минут)**

СЕМЕСТР 6

№ п/п	Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
Тема 5 Неупорядоченные типы данных							
1	Тема 5	ПК-6	ИПК-6.1	2. Задания закрытого типа на установление последовательности	34. Задание: Расположите этапы работы со словарем: 1. Создание словаря 2. Добавление элементов 3. Получение значений 4. Обновление данных Ответ: 1-2-3-4	34. Задание: Расположите операции с множеством в порядке выполнения: 1. Создание множества 2. Добавление элементов 3. Проверка наличия элемента 4. Удаление элемента Ответ: 1-2-3-4	**Базовый (1-3 минуты)**
2	Тема 5	ПК-6	ИПК-6.2	4. Задания комбинированного типа с выбором нескольких ответов	35. Задание: Какие из перечисленных методов используются для работы со словарями? 1. keys() 2. append() 3. values() 4. add() 5. items() Ответ: 1, 3, 5 - методы словарей в Python	35. Задание: Какие операции поддерживают множества? 1. Объединение 2. Пересечение 3. Добавление 4. Сортировка 5. Разность Ответ: 1, 2, 3, 5 - основные операции с множествами	**Базовый (1-3 минуты)**
3	Тема 5	ПК-7	ИПК-7.2	5. Задания открытого типа с развернутым ответом	36. Задание: Создайте словарь для хранения информации о студенте и вычислите средний балл Решение: python student = {"name": "Иван", "age": 20, "grades": [5, 4, 3]} print(f"Средний балл:	36. Задание: Напишите программу для работы с множествами: объединение и пересечение Решение: python set1 = {1, 2, 3, 4, 5} set2 = {4, 5, 6, 7, 8} print(f"Объединение: {set1 | set2}") print(f"Пересечение: {set1 & set2}")	**Высокий (5-10 минут)**

<table>
<tr><th>№ п/п</th><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th>Вариант 1</th><th>Вариант 2</th><th>Уровень освоения компетенции</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>{sum(student['grades'])/len(student['grades'])}")</td><td></td><td></td></tr>
<tr><td>4</td><td>Тема 5</td><td>ПК-6</td><td>ИПК-6.1</td><td>1. Задания закрытого типа на установление соответствия</td><td>37. Задание: Соотнесите операции над множествами с их символами:<table><tr><th>Пункт</th><th>Операция</th><th>Пункт</th><th>Оператор</th></tr><tr><td>А</td><td>Объединение</td><td>1</td><td>&</td></tr><tr><td>Б</td><td>Пересечение</td><td>2</td><td>^</td></tr><tr><td>В</td><td>Разность</td><td>3</td><td>|</td></tr><tr><td>Г</td><td>Симметрическая разность</td><td>4</td><td>-</td></tr></table>Ответ: А-3, Б-1, В-4, Г-2</td><td>37. Задание: Соотнесите методы словарей с их описанием:<table><tr><th>Пункт</th><th>Метод</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>get()</td><td>1</td><td>Получение всех ключей</td></tr><tr><td>Б</td><td>keys()</td><td>2</td><td>Безопасное получение значения</td></tr><tr><td>В</td><td>update()</td><td>3</td><td>Обновление словаря</td></tr></table>Ответ: А-2, Б-1, В-3</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>5</td><td>Тема 5</td><td>ПК-7</td><td>ИПК-7.2</td><td>3. Задания комбинированного типа с выбором одного ответа</td><td>38. Задание: Что вернет {1,2,3} & {3,4,5}?
1. {3}
2. {1,2,3,4,5}
3. {1,2}
Ответ: 1) {3} - пересечение множеств</td><td>38. Задание: Что вернет {1,2,3} - {3,4,5}?
1. {1,2}
2. {4,5}
3. {1,2,4,5}
Ответ: 1) {1,2} - разность множеств</td><td>Повышенный (3-5 минут)</td></tr>
<tr><td colspan="7">Тема 6 Вложенные структуры данных</td><td></td></tr>
<tr><td>6</td><td>Тема 6</td><td>ПК-2</td><td>ИПК-2.1</td><td>2. Задания закрытого типа</td><td>39. Задание: Расположите этапы доступа к вложенному элементу:
А) Получение значения
В) Определение внешнего ключа</td><td>39. Задание: Расположите этапы создания вложенной структуры:
А) Создание внешней структуры
В) Создание внутренней структуры</td><td>Базовый (1-3 минуты)</td></tr>
</table>

№ п/п	Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
				на установление последовательности	C) Проверка существования пути D) Определение внутреннего ключа Ответ: C-B-D-A	C) Добавление данных D) Вложение структур Ответ: B-D-A-C	
7	Тема 6	ПК-2	ИПК-2.2	4. Задания комбинированного типа с выбором нескольких ответов	40. Задание: Какие структуры могут быть вложенными? 1. списки 2. словари 3. кортежи 4. множества 5. строки Ответ: 1, 2, 3 - могут быть вложенными	40. Задание: Какие из следующих структур являются корректными? 1. list = [1, 2, [3, 4]] 2. dict = {'key': {'nested': 'value'}} 3. set = {{1,2}, {3,4}} 4. tuple = (1, (2,3), 4) Ответ: 1, 2, 4 - корректные вложенные структуры	**Повышенный (3-5 минут)**
8	Тема 6	ПК-7	ИПК-7.1	5. Задания открытого типа с развернутым ответом	41. Задание: Создайте вложенную структуру для хранения данных о студентах и получите оценку по математике Решение: python students = [{"name": "Иван", "grades": {"math": 5, "physics": 4}}] print(students[0]["grades"]["math"])	41. Задание: Создайте матрицу 3x3 и получите элемент [1][2] Решение: python matrix = [[1,2,3], [4,5,6], [7,8,9]] print(f"Элемент [1][2]: {matrix[1][2]}")	**Высокий (5-10 минут)**
9	Тема 6	ПК-2	ИПК-2.1	1. Задания закрыто	42. Задание: Соотнесите вложенные структуры с их описанием:	42. Задание: Соотнесите доступ к элементам с примерами:	**Базовый (1-3 минуты)**

<table>
<tr><th>№ п/п</th><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th>Вариант 1</th><th>Вариант 2</th><th>Уровень освоения компетенции</th></tr>
<tr><td></td><td></td><td></td><td></td><td>го типа на установление соответствия</td><td><table><tr><th>Пункт</th><th>Структура данных</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>Список списков</td><td>1</td><td>Таблица с именованными столбцами</td></tr><tr><td>Б</td><td>Словарь словарей</td><td>2</td><td>Матрица/двумерный массив</td></tr><tr><td>В</td><td>Список словарей</td><td>3</td><td>Иерархическая структура данных</td></tr></table>Ответ: А-2, Б-3, В-1</td><td><table><tr><th>Пункт</th><th>Выражение</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>data[0][1]</td><td>1</td><td>Доступ к значению во вложенном словаре</td></tr><tr><td>Б</td><td>data['key1']['key2']</td><td>2</td><td>Доступ к элементу вложенного списка</td></tr><tr><td>В</td><td>data[0]['key']</td><td>3</td><td>Доступ в смешанной структуре</td></tr></table>Ответ: А-2, Б-1, В-3</td><td></td></tr>
<tr><td>10</td><td>Тема 6</td><td>ПК-2</td><td>ИПК-2.2</td><td>3. Задания комбинированного типа с выбором одного ответа</td><td>43. Задание: Как получить доступ к элементу во вложенном словаре data = {'a': {'b': 10}}?
1. data[0]['b']
2. data['a']['b']
3. data.a.b
Ответ: 2) data['a']['b'] - правильный синтаксис</td><td>43. Задание: Что вернет data['users'][1]['name'] для data = {'users': [{'name': 'Anna'}, {'name': 'Ivan'}]}?
1. Anna
2. Ivan
3. Ошибка
Ответ: 2) Ivan - второй элемент списка (индекс 1)</td><td>Повышенный (3-5 минут)</td></tr>
<tr><td colspan="7">Тема 7 Функции и модули</td><td></td></tr>
<tr><td>11</td><td>Тема 7</td><td>ПК-7</td><td>ИПК-7.1</td><td>1. Задания закрытого типа на установление соответствия</td><td>44. Задание: Соотнесите понятия функций с их определениями:<table><tr><th>Пункт</th><th>Термин</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>Параметр</td><td>1</td><td>Переменная в определении функции</td></tr><tr><td>Б</td><td>Аргумент</td><td>2</td><td>Значение, передаваемое в функцию</td></tr><tr><td>В</td><td>Возврат</td><td>3</td><td>Результат работы</td></tr></table></td><td>44. Задание: Соотнесите элементы функции с их описанием:<table><tr><th>Пункт</th><th>Ключевое слово</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>def</td><td>1</td><td>Ключевое слово для определения функции</td></tr><tr><td>Б</td><td>return</td><td>2</td><td>Возврат результата</td></tr><tr><td>В</td><td>lambda</td><td>3</td><td>Анонимная функция</td></tr></table></td><td>Базовый (1-3 минуты)</td></tr>
</table>

№ п/п	Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
					<table><tr><td></td><td></td><td></td><td>функции</td></tr><tr><td>Г</td><td>Область видимости</td><td>4</td><td>Контекст доступности переменных</td></tr></table> Ответ: А-1, Б-2, В-3, Г-4	<table><tr><td>Г</td><td>import</td><td>4</td><td>Импорт модуля</td></tr></table> Ответ: А-1, Б-2, В-3, Г-4	
12	Тема 7	ПК-7	ИПК-7.2	2. Задания закрытого типа на установление последовательности	45. Задание: Расположите этапы работы с функцией: A) Возврат результата B) Вызов функции с аргументами C) Определение функции D) Выполнение тела функции Ответ: C-B-D-A	45. Задание: Расположите этапы импорта модуля: A) Использование функций B) Выполнение кода модуля C) Импорт модуля D) Создание пространства имен Ответ: C-B-D-A	**Базовый (1-3 минуты)**

13	Тема 7	ПК-2	ИПК-2.2	5. Задания открытого типа с развернутым ответом	46. Задание: Напишите функцию для вычисления факториала числа
					Решение:
					python
					def factorial(n):
					  if n == 0: return 1
					  return n * factorial(n-1)
					print(factorial(5))
						46. Задание: Напишите функцию, которая проверяет четность числа
						Решение:
						python
						def is_even(n):
						  return n % 2 == 0
						print(is_even(4))
							Повышенный (3-5 минут)

14	Тема 7	ПК-2	ИПК-2.1	4. Задания комбинированного типа с выбором	47. Задание: Какие утверждения о функциях верны? 1. Функция может возвращать несколько значений 2. Функция может не иметь параметров 3. Функция обязательно должна иметь return 4. Функция может вызывать саму себя	47. Задание: Какие характеристики есть у функций Python? 1. Могут иметь параметры по умолчанию 2. Поддерживают рекурсию 3. Могут быть вложенными 4. Всегда возвращают значение Ответ: 1, 2, 3 - характеристики функций Python	**Базовый (1-3 минуты)**

№ п/п	Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
				нескольких ответов	Ответ: 1, 2, 4 - верные утверждения		
15	Тема 7	ПК-6	ИПК-6.2	3. Задания комбинированного типа с выбором одного ответа	48. Задание: Что такое параметр по умолчанию? 1. Параметр, который нельзя изменить 2. Параметр, который имеет значение, используемое если аргумент не передан 3. Параметр, который всегда должен быть последним Ответ: 2) Параметр со значением по умолчанию	48. Задание: Что такое область видимости? 1. Размер файла с кодом 2. Часть программы, в которой имя ссылается на определенный объект 3. Количество строк в функции Ответ: 2) Контекст доступности переменных	**Повышенный (3-5 минут)**
Тема 8 Файлы							
16	Тема 8	ПК-6	ИПК-6.1	2. Задания закрытого типа на установление последовательности	49. Задание: Расположите этапы работы с файлом: A) Закрытие файла B) Чтение/запись данных C) Открытие файла D) Обработка данных Ответ: C-B-D-A	49. Задание: Расположите этапы чтения данных из файла: A) Обработка данных B) Открытие файла C) Чтение содержимого D) Закрытие файла Ответ: B-C-A-D	**Базовый (1-3 минуты)**
17	Тема 8	ПК-6	ИПК-6.2	4. Задания комбинированн	50. Задание: Какие режимы открытия файлов существуют? 1. "r" 2. "w"	50. Задание: Какие преимущества дает использование with при работе с файлами? 1. Автоматическое закрытие файла 2. Обработка исключений	**Базовый (1-3 минуты)**

<table>
<tr><th>№ п/п</th><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th>Вариант 1</th><th>Вариант 2</th><th>Уровень освоения компетенции</th></tr>
<tr><td></td><td></td><td></td><td></td><td>ого типа с выбором нескольких ответов</td><td>3. "a"
4. "rb"
5. "x"
Ответ: 1, 2, 3, 4, 5 - все основные режимы</td><td>3. Более читаемый код
4. Ускорение работы
Ответ: 1, 2, 3 - преимущества конструкции with</td><td></td></tr>
<tr><td>18</td><td>Тема 8</td><td>ПК-7</td><td>ИПК-7.2</td><td>5. Задания открытого типа с развернутым ответом</td><td>51. Задание: Напишите программу для записи и чтения данных из файла
Решение:
python
with open("data.txt", "w") as f:
f.write("Hello\nWorld")
with open("data.txt", "r") as f:
print(f.read())</td><td>51. Задание: Напишите программу для записи чисел в файл и их последующего чтения
Решение:
python
with open("numbers.txt", "w") as f:
for i in range(1, 6):
f.write(f"{i}\n")
with open("numbers.txt", "r") as f:
print(f.read())</td><td>Высокий (5-10 минут)</td></tr>
<tr><td>19</td><td>Тема 8</td><td>ПК-6</td><td>ИПК-6.1</td><td>1. Задания закрытого типа на установление соответствия</td><td>52. Задание: Соотнесите методы работы с файлами с их описанием:<table><tr><th>Пункт</th><th>Метод</th><th>№</th><th>Описание</th></tr><tr><td>А</td><td>open()</td><td>1</td><td>Чтение одной строки</td></tr><tr><td>Б</td><td>read()</td><td>2</td><td>Открытие файла</td></tr><tr><td>В</td><td>readline()</td><td>3</td><td>Закрытие файла</td></tr><tr><td>Г</td><td>close()</td><td>4</td><td>Чтение всего содержимого</td></tr></table>Ответ: А-2, Б-4, В-1, Г-3</td><td>52. Задание: Соотнесите режимы файлов с их описанием:<table><tr><th>Пункт</th><th>Режим</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>"r"</td><td>1</td><td>Запись с очисткой файла</td></tr><tr><td>Б</td><td>"w"</td><td>2</td><td>Чтение файла</td></tr><tr><td>В</td><td>"a"</td><td>3</td><td>Добавление в конец файла</td></tr></table>Ответ: А-2, Б-1, В-3</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>20</td><td>Тема 8</td><td>ПК-7</td><td>ИПК-7.2</td><td>3. Задания комбин</td><td>53. Задание: Для чего используется конструкция with при работе с файлами?
1. Для автоматического закрытия файла</td><td>53. Задание: Какой режим открытия файла нужен для добавления данных в конец существующего файла?</td><td>Базовый (1-3 минуты)</td></tr>
</table>

№ п/п	Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
				ированного типа с выбором одного ответа	2. Для ускорения работы 3. Для красивого форматирования Ответ: 1) Для автоматического закрытия файла	1. "r" 2. "w" 3. "a" 4. "x" Ответ: 3) "a" - режим добавления	

Критерии оценивания диагностической работы:

Оценка	Критерии оценивания
отлично	от 90 до 100 % правильно выполненных заданий
хорошо	от 70 до 89 % правильно выполненных заданий
удовлетворительно	от 50 до 69 % правильно выполненных заданий
неудовлетворительно	менее 50 % правильно выполненных заданий