

ФГБОУ ВО УНИВЕРСИТЕТ БИОТЕХНОЛОГИЙ
Кафедра информационных технологий и моделирования

Рег. № ПН.03-400/3
«20» ср 2026г.

УТВЕРЖДЕН
на заседании кафедры
Протокол от «16» снв. 2026г. № 5
Заведующий кафедрой
информационных технологий и
моделирования



О.В. Агафонова

(подпись)

ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ

Б1.В.09 Программирование на Python

Шифр и наименование дисциплины

09.03.03 Прикладная информатика

Код и наименование направления подготовки

Прикладная информатика

Направленность (профиль)

Паспорт фонда оценочных средств

№ п/п	Наименование разделов и тем	Код контролируемо й компетенции (или ее части)	Наименование оценочного средства
1.	<i>Введение в программирование на Python</i>	ПК-2 ПК-3	Вопросы для собеседования №1
2.	<i>Основы программирования на языке Python</i>		
2.1	Переменные и основные типы данных	ПК-2 ПК-3	Вопросы для коллоквиума №1
2.2	Коллекции в Python	ПК-2 ПК-3	Вопросы для коллоквиума № 2
2.3	Условные операторы	ПК-2 ПК-3	Вопросы для коллоквиума №3
2.4	Циклы и итерации	ПК-2 ПК-3	Вопросы для коллоквиума №4
2.5	Функции	ПК-2 ПК-3	Разноуровневые задания 3
2.6	Декораторы	ПК-2 ПК-3	
2.7	Работа с файлами	ПК-2 ПК-3	
3.	<i>Объектно-ориентированное программирование в Python</i>		
3.1	Основы ООП	ПК-2 ПК-3	Разноуровневые задания 4
3.2	Наследование и полиморфизм	ПК-2 ПК-3	Вопросы для коллоквиума № 3
4.	<i>Профессиональные методы разработки на Python</i>		
4.1	Разработка проектов	ПК-2 ПК-3	Вопросы для коллоквиума № 4

4.2	Работа с внешними зависимостями и библиотеками	ПК-2 ПК-3	
4.3	Взаимодействие с данными	ПК-2 ПК-3	
4.4	Введение в API	ПК-2 ПК-3	
	Контрольная работа, экзамен	ПК-2 ПК-3	Задание для контрольной работы, вопросы к экзамену

Вопросы для собеседования №1

Тема: «Введение в программирование на Python»

1. Язык программирования Python: история создания, основные парадигмы и сферы применения.
2. Ключевые особенности и преимущества Python по сравнению с компилируемыми языками (C++, Java).
3. Интерпретируемость языка Python: принцип работы интерпретатора, понятие байт-кода и виртуальной машины PVM.
4. Динамическая типизация в Python: понятие, механизмы работы, плюсы и минусы.
5. Управление памятью в Python: сборщик мусора (Garbage Collector) и подсчет ссылок.
6. Объектно-ориентированная природа Python: реализация принципов ООП на уровне синтаксиса языка.
7. Стандартная библиотека Python: обзор основных модулей и их назначение.
8. Управление зависимостями: понятие пакета и модуля, структура импорта (import).
9. Инструменты управления пакетами (pip): установка, обновление и удаление сторонних библиотек.
10. Виртуальные окружения (venv, virtualenv): назначение, принцип изоляции проектов и порядок работы.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума №1
Тема: «Переменные и основные типы данных»

1. Перечислить основные встроенные типы данных Python (числовые, строковый, логический, коллекции) и кратко охарактеризовать каждый из них.
2. Объяснить, что такое переменная в Python, как она связана с объектом в памяти и что означает динамическая типизация.
3. Привести примеры числовых типов данных (int, float, complex) и описать, какие операции над ними поддерживаются.
4. Объяснить назначение строкового типа данных, основные операции над строками (индексация, срезы, конкатенация, повторение).
5. Описать тип bool, его возможные значения и типичные области применения (ветвления, циклы, сравнения).
6. Дать определение списка, указать его основные свойства (изменяемость, упорядоченность, возможность хранения разных типов данных).
7. Дать определение кортежа, указать его основные свойства и отличия от списка (неизменяемость и типичные области применения).
8. Дать определение словаря, пояснить понятия «ключ» и «значение», перечислить основные операции работы со словарём.
9. Дать определение множества, описать его свойства (уникальность элементов, отсутствие порядка) и типичные операции (объединение, пересечение, разность).
10. Объяснить разницу между изменяемыми и неизменяемыми типами данных в Python и привести примеры тех и других (строка, кортеж, список, словарь, множество).

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 2

Тема: «Коллекции в Python»

1. Перечислить основные коллекции Python (список, кортеж, словарь, множество) и указать их ключевые характеристики (упорядоченность, изменяемость, уникальность элементов).
2. Описать свойства списка: индексация, срезы, основные методы (append, pop, insert, remove, sort) и их назначение.
3. Описать свойства кортежа: неизменяемость, использование в качестве ключей словарей, методы (count, index) и типичные сценарии применения.
4. Описать структуру словаря: пары ключ-значение, хэшируемость ключей, методы (get, keys, values, items, pop) и итерация по словарю.
5. Описать свойства множества: уникальность элементов, отсутствие порядка, математические операции (union, intersection, difference, symmetric_difference).
6. Сравнить изменяемые коллекции (list, dict, set) и неизменяемые (tuple, frozenset) по возможностям модификации и использованию в качестве ключей.
7. Объяснить работу comprehensions для списков и словарей: синтаксис, примеры применения и преимущества над циклами.
8. Описать вложенные коллекции: списки в списках, словари в словарях, смешанные структуры данных и доступ к элементам.
9. Указать различия между list и tuple по производительности, памяти и сценариям использования в программировании.
10. Объяснить понятие итерабельности коллекций, роль функции len() и методы итерации (for, enumerate, zip).

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 3

Тема: «Условные операторы»

1. Описать назначение условного оператора `if` и его базовый синтаксис в Python.
2. Объяснить роль операторов `elif` и `else`: последовательность выполнения и типичные сценарии применения.
3. Перечислить операторы сравнения (`==`, `!=`, `>`, `<`, `>=`, `<=`) и привести примеры их использования в условиях.
4. Описать логические операторы (`and`, `or`, `not`): приоритет выполнения, таблицы истинности и короткое замыкание.
5. Объяснить работу тернарного (условного) оператора: синтаксис `value_if_true if condition else value_if_false` и преимущества.
6. Описать вложенные условные конструкции: синтаксис, читаемость кода и рекомендации по рефакторингу.
7. Указать особенности работы с пустыми значениями (`None`, `False`, пустые коллекции) в условиях Python.
8. Объяснить роль оператора `in` для проверки принадлежности (строки, списки, множества) и его эффективность.
9. Описать использование `is` и `is not` для сравнения идентичности объектов и отличия от `==`.
10. Объяснить обработку множественных условий: группировка с помощью скобок, приоритет операций и отладка логических ошибок.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 4

Тема: «Циклы и итерации»

1. Описать назначение цикла `for`: синтаксис, итерация по последовательностям (списки, строки, диапазоны) и базовые примеры.

2. Объяснить цикл while: условие продолжения, риск бесконечного цикла и типичные сценарии применения.
3. Перечислить функции range(): параметры (start, stop, step), варианты использования и назначение в циклах.
4. Описать ключевое слово break: прерывание цикла, вложенные циклы и обработка исключений в итерациях.
5. Объяснить ключевое слово continue: пропуск итерации, влияние на выполнение цикла и примеры оптимизации.
6. Указать роль else в циклах: выполнение при нормальном завершении (без break) и отличия от условного else.
7. Описать вложенные циклы: синтаксис, вычислительная сложность ($O(n^2)$) и практические примеры (таблицы умножения).
8. Объяснить итерацию по словарям: методы keys(), values(), items() и параллельная итерация с enumerate().
9. Описать функцию enumerate(): нумерация элементов, кортежи в цикле и применение для индексации.
10. Указать встроенные итераторы: zip() для параллельной итерации, reversed() и sorted() для изменения порядка.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума №5

Тема: «Функции»

1. Описать назначение функций в Python: повторное использование кода, модульность и базовый синтаксис `def`.
2. Объяснить параметры функций: позиционные, именованные, значения по умолчанию и их порядок объявления.
3. Перечислить `*args` и `**kwargs`: сбор аргументов в кортеж и словарь, распаковка и примеры применения.
4. Указать роль аннотаций типов: синтаксис `->` и `:` для возвращаемого значения и параметров, преимущества.
5. Объяснить ключевое слово `return`: возврат значений, множественные возвраты, отсутствие `return` и `None`.
6. Описать область видимости: локальные, глобальные переменные (`global`), неявные глобальные и LEGB-правило.
7. Указать лямбда-функции: синтаксис `lambda`, однострочные выражения и использование в `map()`, `filter()`.
8. Объяснить декораторы: синтаксис `@decorator`, вложенные функции и модификация поведения без изменения кода.
9. Описать рекурсию: базовый случай, стек вызовов, примеры (факториал, фибоначчи) и ограничения по глубине.
10. Указать функции высшего порядка: передача функций как аргументов, возвращение функций и замыкания (`closures`).

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 6

Тема «Декораторы»

1. Описать назначение декораторов: модификация поведения функций без изменения их кода и синтаксис `@decorator`.
2. Объяснить структуру простого декоратора: функция-обертка (`wrapper`), передача декорируемой функции и `*args, **kwargs`.
3. Указать роль `functools.wraps`: сохранение метаданных оригинальной функции (`__name__`, `__doc__`) и пример использования.
4. Описать цепочку декораторов: порядок применения (снизу вверх), вложенность и влияние последовательности.
5. Объяснить декораторы с параметрами: трехслойная структура (`factory` → `decorator` → `wrapper`) и синтаксис `@decorator(arg)`.
6. Перечислить типичные применения: логирование вызовов, замер времени выполнения, кэширование результатов.
7. Описать сохранение сигнатуры функции: использование `inspect.signature` и `functools.wraps` для introspection.
8. Указать работу с классами: декораторы методов (`@classmethod`, `@staticmethod`), `self` в `wrapper`.
9. Объяснить рекурсивные декораторы: применение нескольких декораторов к одному объекту и порядок выполнения.
10. Описать продвинутые примеры: декораторы с состоянием (замыкания), `retry`-логика и `rate limiting`.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 7

Тема «Работа с файлами»

1. Описать встроенную функцию `open()`: параметры `file`, `mode` (`r`, `w`, `a`, `b`), контекстный менеджер `with`.
2. Объяснить текстовые режимы: чтение (`r`), запись (`w`), добавление (`a`), универсальная кодировка (`encoding='utf-8'`).
3. Указать бинарные режимы: работа с изображениями, исполняемыми файлами (`rb`, `wb`), `BytesIO` и `bytearray`.
4. Описать методы чтения: `read()` (полностью), `readline()` (строка), `readlines()` (список строк) и итерация по файлу.
5. Объяснить методы записи: `write()`, `writelines()`, буферизация, принудительная запись `flush()`.
6. Перечислить поиск позиции: `tell()` (текущая позиция), `seek()` (перемещение), `seek(0)` для начала файла.
7. Описать модуль `os`: пути (`os.path.join`, `os.path.exists`), работа с директориями (`os.listdir`, `os.mkdir`).
8. Указать модуль `pathlib`: объектно-ориентированный подход (`Path()`, `read_text()`, `write_text()`, `iterdir()`).
9. Объяснить обработку исключений: `FileNotFoundError`, `PermissionError`, `OSError` и `try-except-finally`.
10. Описать форматы CSV/JSON: модули `csv`, `json` (`csv.reader`, `json.dump/load`) и сериализация данных.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 8

Тема «Основы ООП»

1. Описать четыре основных принципа ООП: инкапсуляция, наследование, полиморфизм, абстракция и их реализация в Python.
2. Объяснить понятие класса: синтаксис class, тело класса, создание экземпляров и атрибут `__name__`.
3. Указать роль метода `__init__`: инициализация объекта, параметры `self`, передача аргументов при создании экземпляра.
4. Описать атрибуты экземпляра: доступ через `self.attribute`, динамическое добавление, приватные атрибуты (`_`, `__`).
5. Объяснить методы экземпляра: первый параметр `self`, вызов через объект, отличия от статических методов.
6. Перечислить магические методы: `__str__`, `__repr__`, `__len__`, их назначение и переопределение.
7. Описать наследование: синтаксис `class Child(Parent)`, вызов `super().__init__()`, множественное наследование.
8. Указать полиморфизм: переопределение методов родителя, duck typing в Python, интерфейсы через абстрактные классы.
9. Объяснить инкапсуляция: публичные/приватные атрибуты (`__private`), property-декораторы (`@property`, `@attribute.setter`).
10. Описать статические методы (`@staticmethod`) и методы класса (`@classmethod`): параметры `cls`, сценарии применения.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 9

Тема «Наследование и полиморфизм»

1. Описать простое наследование: синтаксис `class Child(Parent)`, доступ к атрибутам родителя через `super()`.
2. Объяснить метод `super()`: вызов методов родителя, поддержка множественного наследования и порядок разрешения MRO.
3. Указать множественное наследование: синтаксис `class Child(Parent1, Parent2)`, алмазная проблема и алгоритм C3-линеаризации.
4. Описать переопределение методов: замена поведения родительского метода, вызов родительского через `super()`.
5. Объяснить полиморфизм: одинаковые имена методов в разных классах, `duck typing` и динамическая диспетчеризация.
6. Перечислить абстрактные классы: модуль `abc`, декоратор `@abstractmethod`, принудительная реализация методов.
7. Указать иерархия классов: построение деревьев наследования, `isinstance()`, `issubclass()` для проверки отношений.
8. Описать метод MRO (Method Resolution Order): алгоритм поиска метода, `Class.__mro__` и отладка конфликтов.
9. Объяснить интерфейсы в Python: протоколы (неявные интерфейсы), структурная типизация и `typing.Protocol`.
10. Указать динамическое наследование: `type()` для создания классов, метаклассы и расширенная настройка поведения.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 10

Тема «Разработка проектов»

1. Описать основные этапы жизненного цикла разработки ПО: планирование, проектирование, разработка, тестирование, деплой, сопровождение.
2. Объяснить назначение структуры проекта: отдельные каталоги для исходного кода, тестов, данных, документации.
3. Перечислить обязательные файлы в корне проекта: README.md, requirements.txt, файл конфигурации, лицензионное соглашение.
4. Описать принципы организации пакетов и модулей: каталог src/app, использование `__init__.py`, группировка по функциональности.
5. Объяснить роль виртуального окружения: изоляция зависимостей проекта, команды создания и активации `venv`.
6. Указать назначение файла requirements.txt или аналогов (pyproject.toml): фиксация версий пакетов и восстановление окружения.
7. Описать использование системы контроля версий (Git): коммиты, ветки, удалённый репозиторий, .gitignore.
8. Объяснить важность тестирования: модульные тесты, каталог tests, базовая структура тестового набора.
9. Перечислить подходы к конфигурации проекта: файлы .env, разделение настроек для dev/test/prod.
10. Описать типичный рабочий цикл разработчика: постановка задачи, реализация, локальное тестирование, коммит, код-ревью, деплой.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 11

Тема «Работа с внешними зависимостями и библиотеками»

1. Объяснить понятие внешней зависимости и библиотеки в Python, отличие стандартной библиотеки от сторонних пакетов.
2. Описать роль менеджера пакетов `pip`: установка, обновление и удаление пакетов, просмотр установленных зависимостей.
3. Объяснить назначение виртуальных окружений (`venv`, `virtualenv`): изоляция зависимостей для разных проектов.
4. Описать структуру и назначение файла `requirements.txt`: фиксация версий, воспроизводимость окружения, команда установки из файла.
5. Перечислить основные источники пакетов: PyPI, локальные и приватные репозитории, зеркала, способы указания индекса в `pip`.
6. Объяснить разницу между зависимостями разработки и боевыми (`prod/dev dependencies`) и способы их разделения.
7. Описать основы управления версиями пакетов: семантическое версионирование, операторы `=`, `>=`, `~=` в требованиях.
8. Указать типичные проблемы несовместимости зависимостей (`dependency hell`) и подходы к их решению.
9. Объяснить назначение современных инструментов управления зависимостями (`pip-tools`, `Poetry`, `Pipenv`) и их отличия от «чистого» `pip`.
10. Описать безопасную работу с внешними библиотеками: проверка источника, обновление версий, отслеживание уязвимостей и лицензий.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 12

Тема «Взаимодействие с данными»

1. Объяснить основные источники данных в проектах: файлы (CSV, JSON), базы данных, сетевые API и пользовательский ввод.
2. Описать типичные форматы данных: текст, CSV, JSON, бинарные форматы; их особенности и области применения.
3. Объяснить базовые операции чтения и записи данных в файлы: последовательное чтение, буферизация, кодировки.
4. Описать принципы сериализации и десериализации данных, назначение модулей json, pickle и их отличия.
5. Объяснить роль структур данных Python (списки, словари) при представлении загруженных данных в памяти.
6. Описать основы работы с табличными данными (строки, столбцы, типы столбцов) на примере CSV.
7. Указать типичные шаги предварительной обработки данных: фильтрация, преобразование типов, удаление пропусков.
8. Объяснить концепцию подключения к внешним источникам данных (БД, API): параметры подключения, конфиденциальность настроек.
9. Описать основы работы с запросами к данным: выборка, сортировка, фильтрация, пагинация результатов.
10. Объяснить важность валидации и проверки качества данных: поиск аномалий, проверка диапазонов и форматов.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 13

Тема «Введение в API»

1. Дать определение API, объяснить назначение программных интерфейсов и их роль во взаимодействии между системами.
2. Объяснить различие между видами API: библиотечные (внутриязыковые), веб-API, системные API, драйверы.
3. Описать основные принципы веб-API: клиент–серверное взаимодействие, запрос–ответ, использование HTTP.
4. Объяснить понятия ресурса и эндпоинта: URL, путь, параметры запроса и их назначение.
5. Перечислить основные HTTP-методы (GET, POST, PUT, DELETE, PATCH) и типичные сценарии их использования.
6. Описать форматы обмена данными в веб-API: JSON, XML, отличие текстовых и бинарных форматов.
7. Объяснить базовые подходы к аутентификации и авторизации при работе с API: API-ключи, токены, OAuth на концептуальном уровне.
8. Описать понятия публичного, приватного и партнёрского API и их различия по доступу и назначению.
9. Объяснить роль документации к API: спецификация эндпоинтов, форматов данных, статусов ответов и ограничений.
10. Описать основные коды ответа HTTP (2xx, 4xx, 5xx) и их значение при работе с веб-API.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Задание для контрольной работы
Тема: «Создание и использование классов»

1. Создайте класс для описания объекта Спортсмен.
2. Класс должен содержать 5 атрибутов (ФИО, год рождения, вид спорта, число участия в соревнованиях, сколько раз занял призовое место).
3. Класс должен содержать 3-4 метода (например, участвует в соревнованиях, занял призовое место и т.д.)
4. Создайте 3 экземпляра класса Спортсмен.
5. Выведите на печать значения атрибутов и вызов методов для каждого экземпляра.
6. Напишите методы для изменения значений атрибутов число участия в соревнованиях и сколько раз занял призовое место способом непосредственного изменения значений.
7. Напишите код для изменения значений атрибутов число участия в соревнованиях и сколько раз занял призовое место способом новых методов, включенных в класс
8. Напишите код для изменения значений атрибутов число участия в соревнованиях и сколько раз занял призовое место способом новых методов с пращением, включенных в класс.

Критерии оценки

Для оценки работы вводится 20 балльная оценочная шкала. На заключительном этапе оценочная шкала переводится в «зачтено» или «не зачтено».

- 1) выполнение работы в соответствии с установленными требованиями к содержанию и оформлению: 16-18б;
- 2) выполнение работы с несущественными ошибками: 10-15б;
- 3) выполнение работы с грубыми ошибками: 1-9б;
- 4) сдача работы в установленные рабочими программами сроки: 2б.

Оценочная шкала для итоговой проверки работы заключается в следующем:

1. Для отметки «Зачтено» необходимо набрать свыше 15 баллов.
2. Для отметки «Не зачтено» - количество баллов от 0 до 15.

Вопросы к экзамену

1. Напишите программу, которая определяет, является ли число четным или нечетным.
2. Объясните разницу между списками и кортежами в Python
3. Реализуйте функцию, которая принимает строку и возвращает количество гласных букв в ней.
4. Что такое словари в Python, и каковы основные методы работы с ними?
5. Напишите программу для вычисления факториала числа с использованием рекурсии.
6. Объясните принцип работы оператора try-except в Python.
7. Создайте класс Книга с атрибутами: название, автор, год издания. Реализуйте метод для вывода информации о книге.
8. Что такое наследование в объектно-ориентированном программировании на Python?
9. Напишите программу, которая считывает файл и выводит количество строк, слов и символов в нем.
10. Как работают списковые включения (list comprehensions) в Python? Приведите пример.
11. Реализуйте функцию, которая принимает список чисел и возвращает список квадратов этих чисел.
12. Объясните, что такое декораторы в Python и для чего они используются.
13. Напишите программу, которая сортирует список слов по длине слова.
14. Что такое модуль в Python? Как импортировать модуль, и в чем разница между import и from ... import?
15. Реализуйте программу, которая использует генератор для создания последовательности простых чисел.
16. Объясните, что такое замыкание в Python, и приведите пример.

17. Напишите скрипт, который считывает аргументы командной строки и выводит их.
18. Объясните основные возможности библиотеки `datetime` в Python.
19. Реализуйте программу, которая создаёт и запускает два потока, выводящих числа от 1 до 10.
20. Что такое GIL (Global Interpreter Lock) в Python, и как он влияет на многопоточность?

Критерии оценки:

Отметка **«ОТЛИЧНО»** - дан полный, развернутый ответ на поставленные вопросы, показана совокупность осознанных знаний по дисциплине; в ответе прослеживается четкая структура и логическая последовательность, отражающая сущность раскрываемых понятий. Ответ изложен литературным языком с использованием современной гистологической терминологии. Могут быть допущены недочеты в определении понятий, исправленные студентом самостоятельно в процессе ответа.

Отметка **«ХОРОШО»** - дан полный, развернутый ответ на поставленный вопросы. Ответ четко структурирован, логичен, изложен литературным языком с использованием современной гистологической терминологии. Могут быть допущены 2-3 неточности или незначительные ошибки, исправленные студентом с помощью преподавателя

Отметка **«УДОВЛЕТВОРИТЕЛЬНО»** - даны недостаточно полный и недостаточно развернутый ответы. Логика и последовательность изложения имеют нарушения. Допущены ошибки в раскрытии понятий, употреблении терминов. В ответе отсутствуют выводы. Умение раскрыть значение обобщенных знаний не показано. Речевое оформление требует поправок, коррекции.

Отметка **«НЕУДОВЛЕТВОРИТЕЛЬНО»** - ответ представляет собой разрозненные знания с существенными ошибками по вопросам. Присутствуют фрагментарность, нелогичность изложения. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная, гистологическая терминология не используется. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента. Или ответ на вопрос полностью отсутствует, или отказ от ответа.

ЗАДАНИЯ ДЛЯ ОЦЕНКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Задания для оценки уровня сформированности компетенции ПК-2

1. С какими типами данных работает Python (выберете неверный вариант)

- a. переменная
- b. словарь
- c. список
- d. строки
- e. совокупность
- f. числа
- g. множества

Ответ: e

2. Какие типы данных относятся к изменяемым типам данных? (множественный выбор)

- a. переменная
- b. словарь
- c. список
- d. строки
- e. числа
- f. множества

Ответ: b,c

3. Какие типы данных относятся к НЕ изменяемым типам данных? (множественный выбор)

- a. переменная
- b. словарь
- c. список
- d. строки
- e. числа
- f. множества

Ответ: a,d,e

4. Тип данных характеризует одновременно:

- a. множество допустимых значений, которые могут принимать данные, принадлежащие к этому типу;
- b. набор операций, которые можно осуществлять над данными, принадлежащими к этому типу;
- c. оба варианта.

Ответ: c

5. Данные какого типа заключаются в одинарные или двойные кавычки?

Ответ: ...

6. Этот тип данных используется для хранения числовых значений.

Ответ: ...

7. Перечислите числовые типы данных.

Ответ: ...

8. Перечислите последовательности, с которыми работает РУ.

Ответ: ...

Задания для оценки уровня сформированности компетенции ПК-3

9. Какие основные компоненты входят в реляционную СУБД?
(выберите неверный вариант)

- a. Таблицы
- b. Связи
- c. Индексы
- d. Триггеры
- e. Нейронные сети
- f. Представления

Ответ: e

10. Какие нормальные формы входят в процесс нормализации базы данных? (множественный выбор)

- a. Первая нормальная форма (1NF)
- b. Вторая нормальная форма (2NF)
- c. Третья нормальная форма (3NF)
- d. Нейронная нормальная форма
- e. Нормальная форма Бойса-Кодда (BCNF)

Ответ: a,b,c,e

11. Какие типы связей существуют между таблицами в реляционной БД? (множественный выбор)

- a. Один-к-одному
- b. Один-ко-многим
- c. Многие-ко-многим
- d. Один-к-нулю
- e. Нелинейная связь

Ответ: a,b,c

12. Верификация структуры БД включает проверку:

- a. Соответствия требованиям заказчика
- b. Наличия первичных ключей во всех таблицах
- c. Отсутствия аномалий вставки, обновления, удаления
- d. Все перечисленные варианты

Ответ: d

13. Данные какого типа используются для первичного ключа таблицы?
Ответ: ...

14. Этот тип объекта БД используется для обеспечения уникальности записей в таблице.
Ответ: ...

15. Перечислите основные ограничения целостности в реляционных БД.
Ответ: ...

16. Перечислите основные SQL-команды для работы со структурой БД.
Ответ: ...

Критерии оценки результатов тестирования:

– оценка «отлично» выставляется студенту, если он отвечает верно на 80-100% вопросов.

– оценка «хорошо», выставляется студенту, если он отвечает верно на 70-79% вопросов.

– оценка «удовлетворительно», выставляется студенту, если он отвечает верно на 60-69% вопросов.

– оценка «неудовлетворительно» выставляется студенту, если он не освоил материал темы, дает менее 60% правильных ответов.

МАТРИЦА СООТВЕТСТВИЯ КРИТЕРИЕВ ОЦЕНКИ УРОВНЮ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Критерии оценки	Уровень сформированности компетенций
Оценка по пятибалльной системе	
«Отлично»	«Высокий уровень»
«Хорошо»	«Повышенный уровень»
«Удовлетворительно»	«Пороговый уровень»
«Неудовлетворительно»	«Не достаточный»
Оценка по системе «зачет – незачет»	
«Зачтено»	«Достаточный»
«Не зачтено»	«Не достаточный»

**Методические материалы, определяющие процедуру оценивания
знаний, умений, навыков и (или) опыта деятельности, характеризующих
этапы формирования компетенций**

1. Положение «О балльно-рейтинговой системе аттестации студентов»: СМК ПНД 08-01-2022, введено приказом от 28.09.2011 №371-О (<http://nsau.edu.ru/file/403>: режим доступа свободный);

2. Положение «О проведении текущего контроля и промежуточной аттестации обучающихся в ФГБОУ ВО Новосибирский ГАУ»: СМК ПНД 77-01-2022, введено в действие приказом от 03.08.2015 №268а-О (<http://nsau.edu.ru/file/104821>: режим доступа свободный).