

Экономический институт
Кафедра экономической кибернетики и прикладной статистики

Директор

« 28
года



Тхор E.C

(наименование профиля подготовки (специальности, магистерской программы); при отсутствии ставится прочерк)

доцент

Велигура А.В.

Заведующий кафедрой экономической кибернетики и прикладной статистики
Велигура А.В.

Луганск 2025 г.

**Комплект оценочных материалов по дисциплине
«Технологии разработки программных продуктов»**

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

1. *Выберите один правильный ответ.*

Какая методология разработки ПО предполагает итеративное выполнение задач с регулярными доработками?

- А) Waterfall
- Б) Agile
- В) Spiral
- Г) V-Model

Правильный ответ: Б

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

2. *Выберите один правильный ответ.*

Какой этап жизненного цикла ПО включает формулировку требований заказчика?

- А) Проектирование
- Б) Тестирование
- В) Сбор требований
- Г) Внедрение

Правильный ответ: В

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

3. *Выберите один правильный ответ.*

Какой инструмент используется для управления задачами в Agile-методологиях?

- А) Jira
- Б) Photoshop
- В) AutoCAD
- Г) MATLAB

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

4. *Выберите один правильный ответ.*

Что описывает SRS (Software Requirements Specification)?

- А) Архитектуру программы
- Б) Требования к программному продукту
- В) План тестирования
- Г) Пользовательский интерфейс

Правильный ответ: Б

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

5. Выберите один правильный ответ.

Какой из перечисленных процессов относится к этапу внедрения ПО?

- А) Написание кода
- Б) Обучение пользователей
- В) Создание UML-диаграмм
- Г) Юнит-тестирование

Правильный ответ: Б

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

6. Выберите один правильный ответ.

Какая технология используется для автоматизации сборки и развертывания ПО?

- А) Git
- Б) Docker
- В) Jenkins
- Г) PyCharm

Правильный ответ: В

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

7. Выберите один правильный ответ.

Что является основным документом при передаче продукта заказчику?

- А) Техническое задание
- Б) Пользовательская документация
- В) Исходный код
- Г) Отчет об ошибках

Правильный ответ: Б

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

8. Выберите один правильный ответ.

Какой метод применяется для оценки трудоемкости задач в проекте?

- А) Парное программирование
- Б) Planning Poker
- В) Рефакторинг
- Г) Модульное тестирование

Правильный ответ: Б

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

9. Выберите один правильный ответ.

Какой этап следует после тестирования в жизненном цикле ПО?

- А) Сбор требований
- Б) Внедрение
- В) Проектирование
- Г) Разработка

Правильный ответ: Б

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

10. Выберите один правильный ответ.

Какая технология позволяет отслеживать изменения в исходном коде?

А) GitHub

Б) Trello

В) Slack

Г) Confluence

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

11. Выберите один правильный ответ

Какой из перечисленных принципов является основным в объектно-ориентированном программировании?

А) Инкапсуляция

Б) Линейное программирование

В) Функциональное программирование

Г) Модульное программирование

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

12. Выберите один правильный ответ.

Какой метод используется для создания экземпляра класса в Python?

А) `__init__`

Б) `__new__`

В) `__call__`

Г) `__class__`

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

13. Выберите один правильный ответ.

Какой из перечисленных элементов является примером наследования в Python?

А) Создание нового класса на основе существующего

Б) Использование глобальных переменных

В) Перегрузка операторов

Г) Использование декораторов

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

14. Выберите один правильный ответ.

Какой из перечисленных методов используется для перегрузки оператора сложения в Python?

А) `__add__`

- Б) `__sum__`
- В) `__plus__`
- Г) `__concat__`

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

15. Выберите один правильный ответ.

Какой из перечисленных элементов является примером полиморфизма в Python?

- А) Возможность использовать один и тот же метод для разных классов
- Б) Использование глобальных переменных
- В) Перегрузка операторов
- Г) Использование декораторов

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

16. Выберите один правильный ответ.

Какой из перечисленных методов используется для ограничения доступа к атрибутам класса в Python?

- А) Использование приватных атрибутов (например, `__attribute`)
- Б) Использование глобальных переменных
- В) Использование декораторов
- Г) Использование статических методов

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

17. Выберите один правильный ответ

Какой из перечисленных методов используется для создания статического метода в Python?

- А) `@staticmethod`
- Б) `@classmethod`
- В) `@property`
- Г) `@decorator`

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

18. Выберите один правильный ответ

Какой из перечисленных методов используется для создания свойства класса в Python?

- А) `@property`
- Б) `@staticmethod`
- В) `@classmethod`
- Г) `@decorator`

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

19. Выберите один правильный ответ.

Какой из перечисленных методов используется для создания итератора в Python?

А) `__iter__` и `__next__`

Б) `__init__` и `__call__`

В) `__add__` и `__sub__`

Г) `__get__` и `__set__`

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

20. Выберите один правильный ответ.

Какой из перечисленных методов используется для создания контекстного менеджера в Python?

А) `__enter__` и `__exit__`

Б) `__init__` и `__del__`

В) `__call__` и `__new__`

Г) `__get__` и `__set__`

Правильный ответ: А

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

21. Выберите все правильные варианты ответов.

Какие из следующих инструментальных средств используются для работы с файлами в Python?

А) `open()`

Б) `close()`

В) `read()`

Г) `write()`

Правильный ответ: А, В

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

22. Выберите все правильные варианты ответов.

Какие из следующих элементов связаны с модулями и функциями в Python?

А) `import`

Б) `from ... import`

В) `sys.modules`

Г) `inspect.getmodule()`

Правильный ответ: А, Б

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

23. Выберите все правильные варианты ответов.

Какие из следующих методов класса обеспечивают работу с атрибутами?

А) `hasattr()`

Б) `setattr()`

В) `delattr()`

Г) `getattr()`

Правильный ответ: А, В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

24. *Выберите все правильные варианты ответов.*

Какие из следующих инструментальных средств используются для работы с циклами в Python?

А) `range()`

Б) `enumerate()`

В) `for`

Г) `while`

Правильные ответы: А, Б, В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

25. *Выберите все правильные варианты ответов.*

Какие из следующих методов используются для работы со строками в Python?

А) `len()`

Б) `format()`

В) `split()`

Г) `find()`

Правильный ответ: Б, В

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

26. *Выберите все правильные варианты ответов.*

Какие из следующих методов используются для работы с экземплярами класса в Python?

А) `__init__`

Б) `self`

В) `print(self.attr)`

Г) `del self.attr`

Правильный ответ: А, Б, В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

27. *Выберите все правильные варианты ответов.*

Какие из следующих инструментальных средств используются для работы с модулями в Python?

А) `import`

Б) `from ... import`

В) `sys.modules`

Г) `inspect.getmodule()`

Правильный ответ: А, Б, В

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

28. *Выберите все правильные варианты ответов.*

Какие из следующих инструментальных средств используются для тестирования ПО в Python?

А) unittest

Б) pytest

В) nose

Г) coverage

Правильный ответ: А, Б, В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

29. Выберите все правильные варианты ответов.

Какие из следующих инструментальных средств используются для работы с документацией в Python?

А) doctest

Б) sphinx

В) reStructuredText

Г) Markdown

Правильный ответ: А, Б, В

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

30. Выберите все правильные варианты ответов

Какие из следующих инструментальных средств используются для разработки графических интерфейсов в Python?

А) tkinter

Б) PyQt5

В) wxPython

Г) Kivy

Правильный ответ: А, Б, В

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

31. Выберите все правильные варианты ответов.

Какие из следующих методов класса используются для работы с атрибутами?

А) hasattr()

Б) setattr()

В) delattr()

Г) getattr()

Правильный ответ: А, Б, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

32. Выберите все правильные варианты ответов.

Какие из следующих ключевых слов используются при наследовании классов в Python?

А) is

Б) hasattr()

В) super()

Г) self

Правильные ответы: В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

33. Выберите все правильные варианты ответов.

Какие из следующих выражений демонстрируют полиморфизм в Python?

А) `isinstance(obj, type)`

Б) `type(obj)`

В) `obj.__class__`

Г) `print(type(obj))`

Правильные ответы: А, В

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

34. Выберите все правильные варианты ответов.

Какие из следующих ключевых слов используются для определения `static` методов?

А) `static`

Б) `class`

В) `self`

Г) `cls`

Правильные ответы: А, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

35. Выберите все правильные варианты ответов.

Какие из следующих ключевых слов указывают на абстрактные методы?

А) `@abstractmethod`

Б) `@staticmethod`

В) `@property`

Г) `@classmethod`

Правильные ответы: А, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

36. Выберите все правильные варианты ответов.

Какие из следующих ключевых слов используются для определения рекурсивных функций?

А) `def`

Б) `return`

В) `global`

Г) `nonlocal`

Правильные ответы: А, В

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

37. Выберите все правильные варианты ответов.

Какие из следующих ключевых слов используются для перегрузки операторов?

А) `__add__`

- Б) `__eq__`
- В) `__str__`
- Г) `__int__`

Правильные ответы: А, В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

38. Выберите все правильные варианты ответов.

Какие из следующих методов используются для работы с экземплярами класса?

- А) `__init__`
- Б) `self`
- В) `print(self.attr)`
- Г) `del self.attr`

Правильные ответы: А, В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

39. Выберите все правильные варианты ответов.

Какие из следующих ключевых слов используются для работы с `static` и `class` методами?

- А) `static`
- Б) `class`
- В) `self`
- Г) `cls`

Правильные ответы: А, Б, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

40. Выберите все правильные варианты ответов.

Какой из следующих примеров демонстрирует полиморфизм?

- А) `if obj is instance of list: print('It's a list')`
- Б) `print(type(obj))`
- В) `obj.to_string()`
- Г) `obj.sort()`

Правильные ответы: В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

Задания закрытого типа на установление соответствия

1. Установите соответствие между этапами разработки ПО и инструментами Python. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Этапы разработки

Инструменты

- | | |
|------------------------------------|---------------------------------|
| 1) Проектирование структуры данных | А) Модуль <code>unittest</code> |
| 2) Реализация алгоритмов | Б) Генераторы списков |
| 3) Тестирование функционала | В) Классы и наследование |

4) Документирование кода

Г) Строки с тройными кавычками
("""")

Правильный ответ: 1-В, 2-Б, 3-А, 4-Г.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

2. Установите соответствие между задачами этапа тестирования и методами Python. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Задачи тестирования

Методы Python

А) Проверка корректности ввода

1) Блок try/except

Б) Автоматизация тестов

2) Декораторы функций

В) Валидация данных

3) Модуль doctest

Г) Логирование ошибок

4) Метод assert

Правильный ответ: А-1, Б-3, В-4, Г-2.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

3. Установите соответствие между этапами работы с файлами и операциями Python. Каждому элементу левого столбца соответствует только один элемент правого столбца

Этапы работы

Операции Python

А) Открытие файла

1) Метод readlines()

Б) Чтение данных

2) Инструкция with open(...)

В) Запись данных

3) Метод write()

Г) Закрытие файла

4) Метод close()

Правильный ответ: А-2, Б-1, В-3, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

4. Установите соответствие между структурами данных и их применением. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Структуры данных

Применение

А) Список

1) Хранение пар «ключ-значение»

Б) Словарь

2) Обработка последовательностей с быстрым доступом

В) Множество

3) Удаление дубликатов

Г) Кортеж

4) Неизменяемые коллекции

Правильный ответ: А-2, Б-1, В-3, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

5. Установите соответствие между методами ООП и их описанием. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Методы ООП

Описание

А) Наследование

1) Создание методов, работающих с классом, а не экземпляром

- | | |
|-----------------|------------------------------------|
| Б) Инкапсуляция | 2) Соккрытие внутренней реализации |
| В) Полиморфизм | 3) Перегрузка операторов |
| Г) Абстракция | 4) Выделение общего интерфейса |

Правильный ответ: А-1, Б-2, В-3, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

6. Установите соответствие между этапами разработки интерфейса и инструментами Python. Каждому элементу левого столбца соответствует только один элемент правого столбца:

- | Этапы разработки | Инструменты |
|----------------------|-----------------------------|
| А) Проектирование UI | 1) Библиотека Tkinter |
| Б) Обработка событий | 2) Модуль sys.argv |
| В) Валидация данных | 3) Декораторы @staticmethod |
| Г) Запуск приложения | 4) Функция mainloop() |

Правильный ответ: А-1, Б-3, В-2, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

7. Установите соответствие между типами тестирования и их целями. Каждому элементу левого столбца соответствует только один элемент правого столбца:

- | Типы тестирования | Цели |
|-------------------|--|
| А) Модульное | 1) Проверка взаимодействия компонентов |
| Б) Интеграционное | 2) Проверка отдельной функции |
| В) Системное | 3) Проверка всего приложения |
| Г) Регрессионное | 4) Поиск ошибок после изменений |

Правильный ответ: А-2, Б-1, В-3, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

8. Установите соответствие между этапами обработки данных и методами Python. Каждому элементу левого столбца соответствует только один элемент правого столбца:

- | Этапы обработки | Методы |
|----------------------|--------------------------|
| А) Извлечение данных | 1) Метод split() |
| Б) Очистка данных | 2) Библиотека re |
| В) Анализ данных | 3) Модуль collections |
| Г) Визуализация | 4) Библиотека matplotlib |

Правильный ответ: А-2, Б-1, В-3, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

9. Установите соответствие между этапами работы с БД и операциями Python. Каждому элементу левого столбца соответствует только один элемент правого столбца:

- | Этапы работы | Операции |
|---------------------|---------------------|
| А) Подключение к БД | 1) Метод fetchall() |

- | | |
|-------------------------|--------------------|
| Б) Выполнение запроса | 2) Модуль sqlite3 |
| В) Получение данных | 3) Метод execute() |
| Г) Заккрытие соединения | 4) Метод close() |

Правильный ответ: А-2, Б-3, В-1, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

10. Установите соответствие между концепциями ООП и их определениями. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Концепции	Определения
А) Наследование	1) Соккрытие внутренней реализации объекта
Б) Инкапсуляция	2) Возможность использовать один интерфейс для разных типов данных
В) Полиморфизм	3) Создание нового класса на основе существующего
Г) Абстракция	4) Выделение общего поведения без деталей реализации

Правильный ответ: А-3, Б-1, В-2, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

11. Установите соответствие между магическими методами Python и их функциями. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Методы	Функции
А) __init__	1) Возвращает строковое представление объекта
Б) __str__	2) Инициализация объекта
В) __add__	3) Определение длины объекта
Г) __len__	4) Перегрузка оператора сложения

Правильный ответ: А-2, Б-1, В-4, Г-3.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

12. Установите соответствие между элементами класса и их описанием. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Элементы	Описание
А) Атрибут класса	1) Метод, работающий с экземпляром
Б) Статический метод	2) Переменная, общая для всех экземпляров
В) Метод экземпляра	3) Метод, не требующий доступа к экземпляру
Г) Свойство (@property)	4) Контроль доступа к атрибуту через геттер/сеттер

Правильный ответ: А-2, Б-3, В-1, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

13. Установите соответствие между примерами кода и концепциями ООП. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Пример кода	Концепция
А) class Dog(Animal): ...	1) Инкапсуляция
Б) self.__password = "123"	2) Наследование
В) def area(self): raise NotImplementedError	3) Абстракция
Г) def __add__(self, other): ...	4) Полиморфизм

Правильный ответ: А-2, Б-1, В-3, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

14. Установите соответствие между задачами и решениями на Python. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Задачи	Решения
А) Запретить изменение атрибута	1) Использовать @property
Б) Создать неизменяемый класс	2) Объявить атрибут через __slots__
В) Динамически добавлять методы	3) Использовать декоратор @staticmethod
Г) Оптимизировать память объектов	4) Наследовать от tuple

Правильный ответ: А-1, Б-4, В-3, Г-2.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

15. Установите соответствие между типами методов и их использованием. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Методы	Использование
А) @classmethod	1) Работа с данными экземпляра
Б) @staticmethod	2) Создание фабричных методов
В) @property	3) Вызов без доступа к классу/экземпляру
Г) Обычный метод	4) Контроль доступа к атрибуту

Правильный ответ: А-2, Б-3, В-4, Г-1.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

16. Установите соответствие между исключениями и их назначением. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Исключения	Назначение
А) TypeError	1) Ошибка доступа к несуществующему атрибуту

- | | |
|------------------------|---------------------------------|
| Б) AttributeError | 2) Некорректный тип данных |
| В) ValueError | 3) Ошибка в логике наследования |
| Г) NotImplementedError | 4) Некорректное значение |

Правильный ответ: А-2, Б-1, В-4, Г-3.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

17. Установите соответствие между паттернами проектирования и их описанием. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Паттерны	Описание
А) Singleton	1) Создание объектов через фабричный метод
Б) Factory	2) Единственный экземпляр класса
В) Decorator	3) Динамическое добавление функциональности
Г) Observer	4) Уведомление зависимых объектов

Правильный ответ: А-2, Б-1, В-3, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

18. Установите соответствие между элементами кода и их результатом. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Код	Результат
А) super().__init__()	1) Вызов конструктора родительского класса
Б) isinstance(obj, MyClass)	2) Проверка наследования
В) isinstance(Dog, Animal)	3) Проверка типа объекта
Г) @abstractmethod	4) Обязательная реализация метода

Правильный ответ: А-1, Б-3, В-2, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

19. Установите соответствие между задачами и инструментами ООП. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Задачи	Инструменты
А) Реализовать интерфейс	1) Абстрактные классы (abc)
Б) Запретить наследование	2) Декоратор @final
В) Создать итератор	3) Методы __iter__ и __next__
Г) Сериализация объекта	4) Модуль pickle

Правильный ответ: А-1, Б-2, В-3, Г-4.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

Задания закрытого типа на установление правильной последовательности

1. *Разработчик создает программу на Python для обработки данных. Установите правильную последовательность этапов. Запишите правильную последовательность букв:*

- А) Написание кода для обработки данных.
- Б) Тестирование программы на тестовых наборах.
- В) Составление технического задания.
- Г) Документирование кода.
- Д) Отладка ошибок.

Правильный ответ: В, А, Г, Б, Д.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

2. *Программист работает с файлами в Python. Установите правильную последовательность операций. Запишите правильную последовательность букв:*

- А) Закрытие файла методом close().
- Б) Чтение данных через read().
- В) Открытие файла с помощью open().
- Г) Запись данных через write().

Правильный ответ: В, Г, Б, А.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

3. *Студент разрабатывает класс для моделирования банковского счета. Установите последовательность действий. Запишите правильную последовательность букв:*

- А) Реализация методов deposit() и withdraw().
- Б) Определение класса BankAccount.
- В) Создание экземпляра класса: account = BankAccount().
- Г) Написание конструктора __init__().

Правильный ответ: Б, Г, А, В.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

4. *Тестировщик проверяет программу. Установите последовательность этапов тестирования. Запишите правильную последовательность букв:*

- А) Написание юнит-тестов.
- Б) Анализ результатов тестирования.
- В) Запуск тестов через unittest.
- Г) Исправление выявленных ошибок.

Правильный ответ: А, В, Б, Г.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

5. *Разработчик использует Git для управления версиями. Установите последовательность команд. Запишите правильную последовательность букв:*

А) `git commit -m "Message"`.

Б) `git add ..`

В) `git push origin main`.

Г) `git pull`.

Правильный ответ: Б, А, В.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

6. Программист создает веб-приложение на Flask. Установите последовательность шагов. Запишите правильную последовательность букв:

А) Настройка маршрутов через `@app.route()`.

Б) Установка Flask через `pip install flask`.

В) Запуск приложения с `app.run()`.

Г) Создание экземпляра приложения: `app = Flask(__name__)`.

Правильный ответ: Б, Г, А, В.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

7. Аналитик обрабатывает данные с помощью Pandas. Установите последовательность операций. Запишите правильную последовательность букв:

А) Загрузка данных через `pd.read_csv()`.

Б) Визуализация данных с `matplotlib`.

В) Очистка данных от пропусков.

Г) Анализ данных через `groupby()`.

Правильный ответ: А, В, Г, Б.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

8. Разработчик настраивает CI/CD-пайплайн. Установите последовательность этапов. Запишите правильную последовательность букв:

А) Автоматическое развертывание на сервере.

Б) Написание скриптов для тестирования.

В) Интеграция с репозиторием GitHub.

Г) Запуск пайплайна при пуше в ветку.

Правильный ответ: В, Б, Г, А.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

9. Программист работает с SQLite в Python. Установите последовательность операций. Запишите правильную последовательность букв:

А) Выполнение запроса через `cursor.execute()`.

Б) Закрытие соединения с БД.

В) Подключение к БД через `sqlite3.connect()`.

Г) Фиксация изменений методом `commit()`.

Правильный ответ: В, А, Г, Б.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

10. Студент реализует алгоритм сортировки. Установите последовательность шагов. Запишите правильную последовательность букв:

А) Написание функции sort().

Б) Проверка алгоритма на тестовых данных.

В) Анализ сложности алгоритма.

Г) Выбор метода сортировки (например, пузырьковая).

Правильный ответ: Г, А, Б, В.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

11. Бизнес-аналитик разрабатывает систему для генерации отчетов о продажах. Необходимо создать базовый класс *Report* с методом *generate()*, а также подкласс *SalesReport*, переопределяющий этот метод. Запишите правильную последовательность букв, определяющую верный порядок строк кода:

Фрагменты кода:

А) `class SalesReport(Report):`

Б) `def generate(self):`
 `return "Общий отчет"`

В) `class Report:`

Г) `def generate(self):`
 `return "Отчет о продажах"`

Правильный ответ: В, Б, А, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

12. Требуется реализовать систему расчета зарплат сотрудников. Базовый класс *Employee* должен иметь метод *calculate_salary()*, а класс *Manager* — добавлять бонусы. Запишите правильную последовательность букв, определяющую верный порядок строк кода:

Фрагменты:

А) `class Manager(Employee):`

Б) `def calculate_salary(self):`
 `return self.base + 1000`

В) `class Employee:`

Г) `def calculate_salary(self):`
 `return self.base`

Правильный ответ: В, Г, А, Б

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

13. Нужно создать систему учета клиентов. Класс *Client* хранит имя и баланс, а класс *VIPClient* добавляет поле *discount*. Запишите правильную последовательность букв, определяющую верный порядок строк кода:

Фрагменты:

А) `class Client:`

Б) `def __init__(self, name, balance):`

```
self.name = name
self.balance = balance
```

В) class VIPClient(Client):

```
Г) def __init__(self, name, balance, discount):
    super().__init__(name, balance)
    self.discount = discount
```

Правильный ответ: А, Б, В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

14. Разработайте классы для анализа финансовых транзакций. Класс *Transaction* содержит сумму и дату, а класс *FraudTransaction* добавляет флаг *is_fraud*. Запишите правильную последовательность букв, определяющую верный порядок строк кода:

Фрагменты:

А) class FraudTransaction(Transaction):

Б) class Transaction:

```
В) def __init__(self, amount, date):
    self.amount = amount
    self.date = date
```

```
Г) def __init__(self, amount, date, is_fraud):
    super().__init__(amount, date)
    self.is_fraud = is_fraud
```

Правильный ответ: Б, В, А, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

15. Создайте иерархию для прогнозирования спроса. Базовый класс *Forecast* имеет метод *predict()*, класс *LinearForecast* переопределяет его. Запишите правильную последовательность букв, определяющую верный порядок строк кода:

Фрагменты:

А) class Forecast:

```
Б) def predict(self):
    return "Базовый прогноз"
```

В) class LinearForecast(Forecast):

```
Г) def predict(self):
    return "Линейная модель"
```

Правильный ответ: А, Б, В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

16. Реализуйте систему учета проектов. Класс *Project* содержит название и бюджет, класс *ExternalProject* добавляет поле *contractor*. Запишите правильную последовательность букв, определяющую верный порядок строк кода:

Фрагменты:

А) class ExternalProject(Project):

Б) class Project:

В) def __init__(self, title, budget, contractor):

 super().__init__(title, budget)

 self.contractor = contractor

Г) def __init__(self, title, budget):

 self.title = title

 self.budget = budget

Правильный ответ: Б, Г, А, В

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

17. *Создайте систему для анализа KPI. Класс KPI имеет метод calculate(), класс SalesKPI добавляет расчет бонусов. Запишите правильную последовательность букв, определяющую верный порядок строк кода:*

Фрагменты:

А) class SalesKPI(KPI):

Б) def calculate(self):

 return self.value * 0.1

В) class KPI:

Г) def calculate(self):

 return self.value

Правильный ответ: В, Г, А, Б

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

18. *Разработайте систему управления складом. Класс Product хранит название и цену, класс DiscountedProduct добавляет скидку. Запишите правильную последовательность букв, определяющую верный порядок строк кода:*

Фрагменты:

А) class DiscountedProduct(Product):

Б) class Product:

В) def __init__(self, name, price, discount):

 super().__init__(name, price)

 self.discount = discount

Г) def __init__(self, name, price):

 self.name = name;

 self.price = price

Правильный ответ: Б, Г, А, В

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

19. *Нужно создать систему для анализа ROI. Базовый класс Investment содержит метод calculate_roi(), класс TechInvestment переопределяет его. Запишите правильную последовательность букв, определяющую верный порядок строк кода:*

Фрагменты:

А) class Investment:

Б) `def calculate_roi(self):`
 `return (self.profit / self.cost) * 100`

В) `class TechInvestment(Investment):`

Г) `def calculate_roi(self):`
 `return super().calculate_roi() + 5`

Правильный ответ: А, Б, В, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

20. Реализуйте систему формирования бюджетов. Класс *Department* хранит название и бюджет, класс *FinanceDepartment* добавляет метод `add_bonus()`. Запишите правильную последовательность букв, определяющую верный порядок строк кода:

Фрагменты:

А) `class FinanceDepartment(Department):`

Б) `class Department:`

В) `def __init__(self, name, budget):`
 `self.name = name`
 `self.budget = budget`

Г) `def add_bonus(self, amount):`
 `self.budget += amount`

Правильный ответ: Б, В, А, Г

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

Задания открытого типа

Задания открытого типа на дополнение

1. Напишите пропущенное слово (словосочетание).

Процесс формализации требований к программному продукту на начальном этапе разработки называется _____ требований.

Правильный ответ: формулирование/анализ

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

2. Напишите пропущенное слово (словосочетание).

Для хранения упорядоченных изменяемых коллекций элементов в Python используется структура данных _____.

Правильный ответ: список

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

3. Напишите пропущенное слово (словосочетание).

Принцип объектно-ориентированного программирования, позволяющий скрыть детали реализации, называется _____.

Правильный ответ: инкапсуляция

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

4. *Напишите пропущенное слово (словосочетание).*

Этап жизненного цикла программного продукта, на котором происходит написание исходного кода, называется _____.

Правильный ответ: реализация

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

5. *Напишите пропущенное слово (словосочетание).*

Для повторного использования кода и организации функциональности в Python применяются _____.

Правильный ответ: модули

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

6. *Напишите пропущенное слово (словосочетание).*

Методология разработки, предполагающая последовательное выполнение этапов без возврата к предыдущим, называется _____ моделью.

Правильный ответ: каскадная

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

7. *Напишите пропущенное слово (словосочетание).*

Процесс проверки соответствия программы установленным требованиям осуществляется на этапе _____.

Правильный ответ: тестирование

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

8. *Напишите пропущенное слово (словосочетание).*

В Python для управления повторяющимися действиями используются операторы _____.

Правильный ответ: цикла

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

9. *Напишите пропущенное слово (словосочетание).*

Структура данных в Python, предназначенная для хранения пар ключ-значение, называется _____.

Правильный ответ: словарь

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

10. *Напишите пропущенное слово (словосочетание).*

Механизм в ООП, позволяющий классу использовать свойства и методы другого класса, называется _____.

Правильный ответ: наследование

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

11. *Напишите пропущенное слово (словосочетание).*

В Python для создания класса используется ключевое слово _____.

Правильный ответ: class

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

12. *Напишите пропущенное слово (словосочетание).*

Метод, который автоматически вызывается при создании экземпляра класса, называется _____.

Правильный ответ: конструктор (или `__init__`)

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

13. *Напишите пропущенное слово (словосочетание).*

Принцип ООП, позволяющий классу-потомку использовать методы и атрибуты класса-родителя, называется _____.

Правильный ответ: наследование

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

14. *Напишите пропущенное слово (словосочетание).*

Для переопределения стандартных операторов (например, `+` или `==`) в классе используются _____.

Правильный ответ: специальные методы/магические методы

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

15. *Напишите пропущенное слово (словосочетание).*

Метод класса, который может быть вызван без создания экземпляра класса, помечается декоратором _____.

Правильный ответ: `@classmethod`

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

16. *Напишите пропущенное слово (словосочетание).*

Свойство, которое позволяет контролировать доступ к атрибутам класса через методы `getter` и `setter`, реализуется с помощью декоратора _____.

Правильный ответ: `@property`

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

17. *Напишите пропущенное слово (словосочетание).*

Класс, который содержит хотя бы один абстрактный метод, называется _____.

Правильный ответ: абстрактным классом

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

18. *Напишите пропущенное слово (словосочетание).*

Для реализации множественного наследования в Python классы-родители перечисляются через _____.

Правильный ответ: запятую (в определении класса)

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

19. *Напишите пропущенное слово (словосочетание).*

Метод, который позволяет экземпляру класса быть итерируемым объектом, называется _____.

Правильный ответ: `__iter__`

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

20. *Напишите пропущенное слово (словосочетание).*

Механизм, позволяющий добавлять новую функциональность к существующему классу без изменения его исходного кода, называется _____.

Правильный ответ: декоратором/примесью

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

Задания открытого типа с кратким свободным ответом

1. *Дайте ответ на вопрос:* «Опишите суть каскадной модели разработки программного обеспечения и ее ключевой недостаток».

Правильный ответ: Каскадная модель предполагает последовательное выполнение этапов (анализ, проектирование, реализация, тестирование) без возврата к предыдущим стадиям. Ее основной недостаток — низкая гибкость при изменении требований на поздних этапах.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

2. *Дайте ответ на вопрос:* «Какие действия включает процесс формулирования требований к программному продукту?»

Правильный ответ: Этот процесс предполагает сбор, анализ и документирование потребностей заказчика, которые затем преобразуются в техническое задание для разработчиков.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

3. *Дайте ответ на вопрос:* «Объясните, как структура данных «словарь» в Python используется в разработке программных продуктов».

Правильный ответ: Словарь (dict) позволяет хранить данные в виде пар «ключ-значение», что упрощает быстрый доступ и управление информацией, например, настройками или конфигурациями программы.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

4. *Дайте ответ на вопрос:* «Опишите роль этапа реализации (кодирования) в жизненном цикле программного продукта».

Правильный ответ: На этапе реализации разработчики пишут исходный код на основе проектной документации, преобразуя алгоритмы и логику в рабочий программный продукт.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

5. *Дайте ответ на вопрос: «Как принцип инкапсуляции способствует повышению надежности программного обеспечения?»*

Правильный ответ: Инкапсуляция скрывает внутреннюю реализацию класса, ограничивая прямой доступ к данным, что снижает риск ошибок и упрощает модификацию кода.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

6. *Дайте ответ на вопрос: «Для каких задач в Python используется модуль math? Приведите пример функции из этого модуля».*

Правильный ответ: Модуль math предоставляет математические функции, такие как вычисление квадратного корня (sqrt()), тригонометрические операции и работа с числами.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

7. *Дайте ответ на вопрос: «Чем кортеж (tuple) отличается от списка (list) в Python?»*

Правильный ответ: Кортеж — неизменяемая структура данных, что делает его подходящим для хранения константных значений, тогда как список позволяет изменять элементы после создания.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

8. *Дайте ответ на вопрос: «Какие цели преследует этап тестирования программного продукта?»*

Правильный ответ: Тестирование выявляет ошибки в коде, проверяет соответствие программы требованиям заказчика и обеспечивает ее корректную работу в различных сценариях.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

9. *Дайте ответ на вопрос: «Как наследование в ООП упрощает разработку сложных программных систем?»*

Правильный ответ: Наследование позволяет повторно использовать код родительского класса, сокращая дублирование и упрощая поддержку иерархических структур.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

10. *Дайте ответ на вопрос: «Как метод format() в Python улучшает читаемость кода при работе со строками?»*

Правильный ответ: Метод format() позволяет вставлять значения переменных в строку по именованным или позиционным параметрам, делая код более понятным и структурированным.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

11. *Дайте ответ на вопрос: «Какие задачи решает конструктор класса в Python и как он объявляется?»*

Правильный ответ: Конструктор класса (`__init__`) инициализирует атрибуты объекта при его создании. Он объявляется как метод с именем `__init__`, принимающий параметр `self` и другие аргументы.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

12. *Дайте ответ на вопрос: «Как реализуется наследование в Python и какой синтаксис используется?»*

Правильный ответ: Наследование реализуется путем указания родительского класса в скобках при объявлении дочернего класса:

```
class ДочернийКласс(РодительскийКласс)
```

Это позволяет использовать методы и атрибуты родителя.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

13. *Дайте ответ на вопрос: «Для чего применяется декоратор `@property` в классах Python?»*

Правильный ответ: Декоратор `@property` позволяет превратить метод класса в атрибут, что дает контроль над чтением, записью и удалением данных, обеспечивая инкапсуляцию.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

14. *Дайте ответ на вопрос: «Чем отличается статический метод (`@staticmethod`) от метода класса (`@classmethod`)?»*

Правильный ответ: Статический метод не принимает `self` или `cls` и работает как обычная функция внутри класса. Метод класса принимает `cls` и может модифицировать состояние класса.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

15. *Дайте ответ на вопрос: «Как перегрузить оператор сложения (+) для пользовательского класса в Python?»*

Правильный ответ: Для этого определяется метод `__add__(self, other)`, который возвращает новый объект с результатом операции. Например:

```
def __add__(self, other): return self.value + other.value.
```

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

16. *Дайте ответ на вопрос: «Что такое абстрактный класс и как его создать в Python?»*

Правильный ответ: Абстрактный класс содержит хотя бы один абстрактный метод (объявленный с `@abstractmethod`). Для его создания используется модуль `abc` и наследование от `ABC`.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

17. *Дайте ответ на вопрос: «Как реализовать инкапсуляцию в Python, если нет модификаторов доступа?»*

Правильный ответ: Инкапсуляция достигается через соглашение об именовании: атрибуты с одним подчеркиванием (`_name`) считаются защищенными, с двумя (`__name`) – приватными.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

18. *Дайте ответ на вопрос: «Что такое примеси (mixins) и как их применяют в Python»?*

Правильный ответ: Примеси – это классы с определенной функциональностью, которые подключаются через множественное наследование для добавления методов дочерним классам без изменения их основной логики.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

19. *Дайте ответ на вопрос: «Как создать итератор для пользовательского класса в Python»?*

Правильный ответ: Нужно реализовать методы `__iter__()` (возвращает сам объект) и `__next__()` (возвращает следующий элемент или вызывает `StopIteration`).

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

20. *Дайте ответ на вопрос: «Какие преимущества дает использование полиморфизма в ООП»?*

Правильный ответ: Полиморфизм позволяет использовать методы с одинаковым именем в разных классах, что упрощает взаимодействие с объектами различных типов через единый интерфейс.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

Задания открытого типа с развернутым ответом

1. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Опишите этапы жизненного цикла программного продукта и их взаимосвязь.

Время выполнения: 15 минут

Ожидаемый результат:

Основные этапы:

1. Анализ требований — сбор и формализация потребностей заказчика.
2. Проектирование — создание архитектуры системы.
3. Реализация — написание кода на основе проектной документации.
4. Тестирование — проверка на соответствие требованиям.
5. Внедрение — установка и настройка продукта.
6. Сопровождение — исправление ошибок и обновления.

Взаимосвязь: Ошибки на этапе анализа (например, неучтенные требования) приводят к переработке проектирования и кода.

Пример: Если на этапе тестирования выявлены недочеты в логике, требуется вернуться к этапу реализации.

Критерии оценивания: Наличие минимум 4 этапов и пояснение их взаимосвязи.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

2. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Объясните роль инкапсуляции в разработке программных продуктов на Python.

Время выполнения: 10 минут

Ожидаемый результат:

Инкапсуляция – принцип ООП, скрывающий внутреннюю реализацию класса. В Python для этого используются:

- Приватные атрибуты:

```
class BankAccount:
```

```
    def __init__(self):
```

```
        self.__balance = 0 # Приватный атрибут
```

```
    def deposit(self, amount):
```

```
        self.__balance += amount
```

Преимущества:

- Защита данных от случайного изменения.
- Упрощение модификации кода (изменения внутри класса не затрагивают внешний код).

Критерии оценивания: Четкое определение, пример и обоснование значимости.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

3. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Сравните Agile и Waterfall методологии. В каком случае Agile предпочтительнее?

Время выполнения: 20 минут

Ожидаемый результат:

Agile:

- Итеративная разработка, гибкие требования.
- Подходит для проектов с неопределенными требованиями (например, стартапы).

Waterfall:

- Последовательные этапы без возврата.
- Эффективен для проектов с фиксированными требованиями (например, государственные системы).

Пример: Для стартапа, где требования меняются каждую неделю, Agile позволяет быстро адаптироваться.

Риски Waterfall: Задержки из-за невозможности внести изменения после завершения этапа.

Критерии оценивания: Сравнение двух методологий и пример их применения.
Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

4. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Опишите виды тестирования программного продукта и их цели.

Время выполнения: 15 минут

Ожидаемый результат:

1. Модульное: Проверка отдельных компонентов.

```
def test_add():
```

```
    assert add(2, 3) == 5
```

2. Интеграционное: Проверка взаимодействия модулей.

3. Системное: Проверка всей системы на соответствие требованиям.

4. Приемочное: Финальная проверка заказчиком.

Цель: Выявление ошибок, обеспечение качества и соответствия требованиям.

Критерии оценивания: Упоминание минимум 3 видов тестирования и пример.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

5. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Как списки и словари в Python упрощают разработку программных продуктов?

Время выполнения: 10 минут

Ожидаемый результат:

- Списки:

```
orders = ["order1", "order2"] # Упорядоченное хранение данных.
```

- Словари:

```
config = {"port": 8080, "debug": True} # Быстрый доступ по ключу.
```

Сравнение: Списки подходят для итераций, словари — для поиска и управления настройками.

Критерии оценивания: Наличие двух примеров и обоснование выбора.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

6. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Объясните применение регулярных выражений в Python на примере валидации email.

Время выполнения: 15 минут

Ожидаемый результат:

Код:

```
import re
```

```
email = "user@example.com"
```

```
pattern = r"^[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$"
```

```
if re.match(pattern, email):
```

```
    print("Valid")
```

Применение:

- Валидация входных данных в формах.
- Парсинг текстовых логов.

Критерии оценивания: Наличие примера кода и описание минимум 2 сфер применения.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

7. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Какие инструменты Python используются для работы с файлами и каталогами?

Время выполнения: 10 минут

Ожидаемый результат:

- Модуль os: Управление путями, создание папок.
- Модуль shutil: Копирование и удаление файлов.

Пример:

try:

```
with open("data.txt", "r") as file:
    data = file.read()
```

except FileNotFoundError:

```
    print("Файл не найден")
```

Критерии оценивания: Упоминание двух модулей и пример работы с файлом.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

8. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Как наследование способствует повторному использованию кода?

Время выполнения: 10 минут

Ожидаемый результат:

Правильный ответ:

Пример:

```
class Animal:
```

```
    def eat(self):
        print("Eating...")
```

```
class Dog(Animal):
```

```
    def bark(self):
        print("Woof!")
```

Преимущества:

- Сокращение дублирования кода.

Упрощение поддержки иерархии объектов.

Критерии оценивания: Пример иерархии классов и пояснение выгод.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

9. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Опишите этапы внедрения программного продукта в эксплуатацию.

Время выполнения: 15 минут

Ожидаемый результат:

Этапы:

1. Установка на сервера заказчика.
2. Настройка под инфраструктуру.
3. Обучение пользователей.
4. Миграция данных из старых систем.

Риски:

- Ошибки при переносе данных.
- Недостаточная документация для настройки.

Критерии оценивания: Упоминание минимум 3 этапов и пример риска.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

10. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Как методы форматирования строк (format() и f-строки) улучшают читаемость кода?

Время выполнения: 10 минут

Ожидаемый результат:

Пример f-строк:

```
python
```

```
Сору
```

```
name = "Иван"
```

```
print(f"Привет, {name}!") # Читаемо и лаконично.
```

Преимущества:

- Упрощение сложных строковых конструкций.
- Снижение ошибок при конкатенации.

Критерии оценивания: Пример использования и обоснование выбора метода.

Компетенции (индикаторы): ОПК-3 (ОПК-3.1)

11. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Объясните, как создать класс в Python, инициализировать его атрибуты и методы. Приведите пример класса Car с атрибутами model и year, а также методом display_info().

Время выполнения: 10 минут

Ожидаемый результат:

```
class Car:
```

```
    def __init__(self, model, year):
```

```
        self.model = model
```

```
        self.year = year
```

```
    def display_info(self):
```

```
        print(f"Модель: {self.model}, Год выпуска: {self.year}")
```

Пример использования

```
my_car = Car("Toyota Camry", 2020)
```

```
my_car.display_info()
```

Пояснение: Класс Car инкапсулирует данные об автомобиле, а метод display_info предоставляет интерфейс для вывода информации.

Критерии оценивания: Корректный синтаксис, реализация метода, пояснение.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

12. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Создайте класс ElectricCar, который наследует от класса Car (из предыдущего вопроса) и добавляет атрибут battery_capacity. Переопределите метод display_info(), чтобы выводить информацию о батарее.

Время выполнения: 15 минут

Ожидаемый результат:

```
class ElectricCar(Car):
```

```
    def __init__(self, model, year, battery_capacity):
```

```
        super().__init__(model, year)
```

```
        self.battery_capacity = battery_capacity
```

```
    def display_info(self):
```

```
        print(f'Модель: {self.model}, Год: {self.year}, Батарея: {self.battery_capacity} кВт·ч')
```

Пример использования

```
tesla = ElectricCar("Tesla Model S", 2023, 100)
```

```
tesla.display_info()
```

Пояснение: Наследование позволяет повторно использовать код класса Car, добавляя новую функциональность.

Критерии оценивания: Корректное наследование, расширение функциональности.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

13. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Как реализовать инкапсуляцию в Python? Создайте класс BankAccount с приватным атрибутом __balance и методами для внесения (deposit) и снятия (withdraw) средств.

Время выполнения: 15 минут

Ожидаемый результат:

```
class BankAccount:
```

```
    def __init__(self):
```

```
        self.__balance = 0
```

```
    def deposit(self, amount):
```

```

self.__balance += amount

def withdraw(self, amount):
    if amount > self.__balance:
        raise ValueError("Недостаточно средств")
    self.__balance -= amount

def get_balance(self):
    return self.__balance

# Пример использования
account = BankAccount()
account.deposit(1000)
account.withdraw(500)
print(account.get_balance()) # 500

```

Пояснение: Приватные атрибуты и методы управления ими обеспечивают безопасность данных.

Критерии оценивания: Реализация защиты данных, обработки исключений.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

14. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Объясните, как полиморфизм реализуется в Python. Приведите пример с классами Cat и Dog, где оба имеют метод speak(), но выводят разные звуки.

Время выполнения: 10 минут

Ожидаемый результат:

```

class Cat:
    def speak(self):
        print("Мяу!")

```

```

class Dog:
    def speak(self):
        print("Гав!")

```

```

# Полиморфизм
animals = [Cat(), Dog()]
for animal in animals:
    animal.speak()

```

Критерии оценивания: Понимание полиморфизма, корректные примеры.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

15. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Создайте абстрактный класс Shape с методом area(). Реализуйте классы Circle и Rectangle, наследующие Shape.

Время выполнения: 20 минут

Ожидаемый результат:

```
from abc import ABC, abstractmethod
```

```
class Shape(ABC):
```

```
    @abstractmethod
```

```
    def area(self):
```

```
        pass
```

```
class Circle(Shape):
```

```
    def __init__(self, radius):
```

```
        self.radius = radius
```

```
    def area(self):
```

```
        return 3.14 * self.radius ** 2
```

```
class Rectangle(Shape):
```

```
    def __init__(self, width, height):
```

```
        self.width = width
```

```
        self.height = height
```

```
    def area(self):
```

```
        return self.width * self.height
```

```
# Пример использования
```

```
circle = Circle(5)
```

```
print(circle.area()) # 78.5
```

Пояснение: Абстрактные классы задают интерфейс для наследников.

Критерии оценивания: Корректное наследование, реализация методов.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

16. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Для чего используется декоратор @classmethod? Приведите пример класса Student с методом from_csv, который создает объект из строки CSV.

Время выполнения: 15 минут

Ожидаемый результат:

```
class Student:
```

```
    def __init__(self, name, age):
```

```
        self.name = name
```

```
        self.age = age
```

```
    @classmethod
```

```
    def from_csv(cls, csv_string):
```

```
        name, age = csv_string.split(",")
```

```
return cls(name, int(age))
```

Пример использования

```
student = Student.from_csv("Иван,20")
```

```
print(student.name, student.age) # Иван 20
```

Пояснение: @classmethod позволяет создавать объекты альтернативными способами.

Критерии оценивания: Понимание роли @classmethod, рабочий пример.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

17. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Перегрузите оператор + для класса Vector, чтобы он складывал векторы.

Пример: Vector(1, 2) + Vector(3, 4) = Vector(4, 6).

Время выполнения: 15 минут

Ожидаемый результат:

```
class Vector:
```

```
    def __init__(self, x, y):
```

```
        self.x = x
```

```
        self.y = y
```

```
    def __add__(self, other):
```

```
        return Vector(self.x + other.x, self.y + other.y)
```

```
v1 = Vector(1, 2)
```

```
v2 = Vector(3, 4)
```

```
result = v1 + v2
```

```
print(result.x, result.y) # 4 6
```

Пояснение: Магические методы позволяют переопределять поведение операторов.

18. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Что такое миксины и как их использовать в Python? Создайте миксин LoggingMixin, добавляющий метод log для вывода сообщений.

Время выполнения: 20 минут

Ожидаемый результат:

```
class LoggingMixin:
```

```
    def log(self, message):
```

```
        print(f'[LOG] {message}')
```

```
class User(LoggingMixin):
```

```
    def __init__(self, name):
```

```
        self.name = name
```

```
def greet(self):
    self.log(f"Пользователь {self.name} авторизован")
```

```
user = User("Анна")
```

```
user.greet() # [LOG] Пользователь Анна авторизован
```

Пояснение: Миксины добавляют функциональность без изменения основной логики класса.

Критерии оценивания: Корректное наследование, рабочая функциональность.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

19. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Создайте класс Counter, который генерирует числа от start до end с шагом step.

Реализуйте протокол итератора.

Время выполнения: 15 минут

Ожидаемый результат:

```
class Counter:
    def __init__(self, start, end, step):
        self.current = start
        self.end = end
        self.step = step

    def __iter__(self):
        return self

    def __next__(self):
        if self.current >= self.end:
            raise StopIteration
        value = self.current
        self.current += self.step
        return value
```

Пример использования

```
for num in Counter(0, 5, 1):
```

```
    print(num) # 0, 1, 2, 3, 4
```

Пояснение: Итераторы позволяют обходить коллекции или генерировать последовательности.

Критерии оценивания: Корректная реализация итератора.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

20. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Реализуйте паттерн Singleton на Python, чтобы класс Database мог иметь только один экземпляр.

Время выполнения: 20 минут

Ожидаемый результат:

```
class Database:
    __instance = None

    def __new__(cls):
        if cls.__instance is None:
            cls.__instance = super().__new__(cls)
        return cls.__instance
```

```
db1 = Database()
db2 = Database()
print(db1 is db2) # True
```

Пояснение: Паттерн Singleton гарантирует единственный экземпляр класса.

Критерии оценивания: Корректная реализация Singleton.

Компетенции (индикаторы): ОПК-3 (ОПК-3.3)

Экспертное заключение

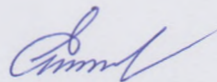
Представленный комплект оценочных материалов по дисциплине (практике) «Технологии разработки программных продуктов» соответствует требованиям ФГОС ВО.

Предлагаемые оценочные материалы адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 38.03.05 Бизнес-информатика.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанные и представленные для экспертизы оценочные материалы рекомендуются к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической
комиссии экономического института



Е.Н. Шаповалова

Лист изменения и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр, на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)