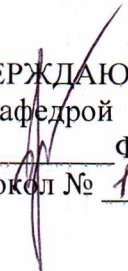


Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»
Рыбницкий филиал

Кафедра автоматизации технологических процессов и производств

УТВЕРЖДАЮ

Зав. кафедрой

 Федоров В.Е., доцент
Протокол № 1 от «14» сентября 2024 г.

Фонд оценочных средств

по дисциплине

«ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ПРОГРАММЫ»

Направление подготовки:

2.15.03.04 «Автоматизация технологических процессов и производств»

Профиль подготовки:

«Автоматизация технологических процессов и производств»

Квалификация

Бакалавр

Форма обучения

очная

2022 год набора

Разработал: доцент  О.В. Шестопал
«14» сентября 2024 г.

Паспорт фонда оценочных средств по учебной дисциплине

1. В результате изучения дисциплины «Объектно-ориентированные программы» у обучающихся должны быть сформированы следующие компетенции:

Категория (группа) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Общепрофесси ональные	ОПК-12. Способен оформлять, представлять и докладывать результаты выполненной работы	ИД _{ОПК-12.1} Готовит аналитические обзоры и делает доклады по результатам выполненной работы ИД _{ОПК-12.2} Подготавливает технические отчеты и публикации по результатам выполненной работы ИД _{ОПК-12.3} Использует современные информационные технологии при подготовке отчетов и заключений по результатам выполненной работы
	ОПК-14. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.	ИД _{ОПК-14.1} Работает с современными информационно-техническими системами и технологиями, используемыми в профильном виде производственной деятельности ИД _{ОПК-14.2} Обеспечивает информационно-техническое сопровождение профессиональной производственной деятельности ИД _{ОПК-14.3} Применяет современные методы исследования и цифровые системы автоматизированного проектирования для изготовления деталей и узлов на станках с числовым программным управлением

2. Программа оценивания контролируемой компетенции:

Текущая аттестация	Контролируемые модули, разделы (темы) дисциплины и их наименование	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Типы и структуры данных	ОПК-12, ОПК-14	Практические и лабораторные работы
2	Основы объектно-ориентированного программирования	ОПК-12, ОПК-14	Практические и лабораторные работы
Промежуточная аттестация		Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1		ОПК-12, ОПК-14	Вопросы к экзамену, тест

«УТВЕРЖДАЮ»

зав. кафедрой АТПиП, доцент

В.Е. Федоров

« 14 »

09

2024 г.

**Вопросы к экзамену
по дисциплине «Объектно-ориентированные программы»
для студентов III курса
направления «Автоматизация технологических процессов и производств»,
профиля «Автоматизация технологических процессов и производств»
V семестр**

1. Эволюция технологий программирования от алгоритмического к ООП. Преимущества ООП.
2. Концепции ООП: инкапсуляция, наследование, полиморфизм.
3. Понятие класса. Объявление и определение элементов класса. Управление доступом к элементам класса. Создание объекта (экземпляра) класса.
4. Назначение конструкторов и деструкторов. Перегрузка конструкторов.
5. Виды конструкторов (конструктор копирования, конструктор по умолчанию).
6. Динамические структуры данных. Указатели. Указатель this.
7. Статические члены: функции и данные. Указатели на члены.
8. Понятие наследования, его назначение. Базовые и производные классы, синтаксис определения производных классов.
9. Понятие полиморфизма. Виртуальные методы: назначение, правила использования.
10. Чисто виртуальные методы и абстрактные классы.
11. Размещение объектов в динамически распределяемой памяти, доступ к членам класса.
12. Динамическое размещение членов класса. Использование указателей, как средство хранения адреса.
13. Ссылки. Передача аргументов функции как ссылок.
14. Пространства имен. Пространства имен, создание и использование, стандартное пространство имен.

Составитель _____ О. В. Шестопал, доцент

Государственное образовательное учреждение
«Приднестровский государственный университет имени Т.Г. Шевченко»
Рыбницкий филиал
Кафедра «Автоматизации технологических процессов и производств»

Задачи к билетам

1. Объявите класс Point3D для точек с тремя координатами x , y , z . Создайте несколько экземпляров этого класса и через них выведите в консоль значения x, y, z . Далее, сделайте следующие манипуляции:

- поменяйте любое значение координаты в классе Point3D и посмотрите, как это повлияет на отображаемые величины экземпляров класса;
- удалите координату z в классе Point3D и убедитесь, что она будет отсутствовать во всех экземплярах;
- поменяйте координату в каком-либо экземпляре класса и посмотрите на результат.

2. Создайте класс Point3D, который хранит координаты в виде списка. Пропишите у него конструктор для создания экземпляров с локальными координатами. Также добавьте методы, позволяющие изменять координаты и получать их (в виде кортежа).

3. Объявите класс Point с конструктором, который бы позволял создавать экземпляр на основе другого, уже существующего. Если аргументы в конструктор не передаются, то создается объект с локальными атрибутами по умолчанию.

4. Напишите программу, в которой пользователь вводит координаты x , y с клавиатуры, создается соответствующий экземпляр, и он сохраняется в списке. Количество вводимых объектов $N=5$. Затем, вывести их атрибуты в консоль.

5. Создайте свою структуру данных, которая имитирует встроенные коллекции

6. Создайте суперкласс «Персональные компьютеры» и на его основе подклассы: «Настольные ПК» и «Ноутбуки». В базовом классе определите общие свойства: размер памяти, диска, модель, CPU. А в производных классах уникальные свойства:

- для настольных: монитор, клавиатура, мышь, их габариты; и метод для вывода этой информации в консоль;
- для ноутбуков: габариты, диагональ экрана; и метод для вывода этой информации в консоль.

7. Напишите класс «Станки» с «полной инкапсуляцией», доступ к атрибутам которого и изменение данных реализуются через вызовы методов `get` и `set`. В объектно-ориентированном программировании принято имена методов для извлечения данных начинать со слова `get` (взять), а имена методов, в которых свойствам присваиваются значения, – со слова `set` (установить). Например, `getField`, `setField`.

8. Напишите класс «Авто» с «полной инкапсуляцией», с доступом через аннотацию свойств и проверкой условий правильности вводимых данных.

9. Написать класс «Стол», который принимает параметры: высота, ширина, высота, тип (круглый или квадратный). Написать два класса «круглый стол» и «прямоугольный стол» наследующих основной класс «Стол». И добавить в новые классы методы расчёта площади столешницы по соответствующим формулам.

10. Написать один класс «Стол» включающим поле «форма столешницы» с методом «площадь столешницы», который вычисляет площадь в зависимости от выбранной формы.

Доцент _____
(подпись)

Шестопад О.В.
(ФИО)

« 17 » 09 _____ 2024 г.

**Государственное образовательное учреждение
«Приднестровский государственный университет имени Т.Г. Шевченко»
Рыбницкий филиал**

Кафедра «Автоматизация технологических процессов и производств»

Тест

по дисциплине «Объектно-ориентированные программы»
(наименование дисциплины)

Указания: Выберите правильное утверждение.

Количество заданий – 10

Время тестирования – 7 минут

1. Какой метод отвечает за добавление элемента в конец списка?

- ☐ add()
- ☐ append()
- ☐ include()
- ☐ extend()

2. Что нужно сделать для того, чтобы воспользоваться созданной реализацией?

- ☐ вызвать имеющиеся методы напрямую
- ☐ создать экземпляр
- ☐ вызвать конструктор по умолчанию
- ☐ создать класс для вызова

3. Какой оператор отвечает за прерывание работы цикла в Python?

- ☐ try-except
- ☐ continue
- ☐ break
- ☐ stop

4. Какое ключевое слово используется в Python для создания класса?

- ☐ declare
- ☐ class
- ☐ create
- ☐ main

5. Какие типы данных допустимы в множестве?

- ☐ строковые и целочисленные
- ☐ числовые
- ☐ любые
- ☐ числовые и строковые

6. Что из предложенного можно считать описание реализации общей задачи в ООП?

- ☐ класс
- ☐ метод
- ☐ функция
- ☐ процедура

7. Какой метод используется для замены подстроки?

- ☐ string.change()
- ☐ string.replace()
- ☐ string.renew()
- ☐ string.update()

8. Что представляет собой Python-список?

- ☐ многомерный массив
- ☐ позиционно упорядоченную коллекцию с произвольными типами элементов
- ☐ упорядоченный массив
- ☐ разновидность FIFO-очереди

9. Какое ключевое слово используется в Python для создания конструктора?

- ☐ __class__
- ☐ __main__
- ☐ __init__
- ☐ __name__

10. Какой элемент отвечает за инициализацию полей класса?

- ☐ интерфейс
- ☐ метод
- ☐ объект
- ☐ конструктор

Данный тест отражает минимальный уровень знаний по дисциплине и оценивается на «удовлетворительно» при не менее 8 верных ответов.

Тест
по дисциплине «Объектно-ориентированные программы»
(наименование дисциплины)

Указания: Выберите правильное утверждение.

Количество заданий – 38

Время тестирования – 30 минут

1. Что такое ООП?

- а) операторно-ориентированное программирование,
- б) основы объектного программирования,
- в) оптимизационно-ориентированное программирование,
- г) объектно-ориентированное программирование.

2. Как называется механизм ООП, который позволяет объединять данные и методы, работающие с этими данными, в единый объект, скрывая при этом детали реализации от пользователя?

- а) полиморфизм, б) инкапсуляция, в) наследование, г) композиция.

3. Какое ключевое слово используется в Python для определения классов?

- а) class, б) cls, в) object, г) obj.

4. Какой стиль именования классов рекомендуется в руководстве PEP 8 по оформлению кода на Python?

- а) camelCase, б) CapWords, в) snake_case, г) ALL_CAPS

5. Какой оператор используется для доступа к атрибуту объекта или класса в Python?

- а) ->, б) ::, в) ., г) [].

6. Укажите специальный метод, который используется для инициализации исходными данными вновь создаваемых экземпляров класса.

- а) __construct__, б) __new__, в) __init__, г) __start__.

7. Что будет выведено на экран в результате выполнения кода условия?

- а) 1, б) 2, в) 3, г) ошибка.

```
class Test:
    def __init__(self, x=1):
        self.x = x
```

```
t_1 = Test()
t_2 = Test(2)
print(t_1.x + t_2.x)
```

8. Что будет выведено на экран в результате выполнения кода условия?

- а) 25, б) 30, в) 55, г) ошибка.

Условие

```
class Person:
    def set_age(self, age):
        self.age = age
```

```
ivan = Person(25)
ivan.set_age(30)
print(ivan.age)
```

9. Перечислите допустимые способы изменения значения атрибута объекта в Python.

- а) obj.attr = value, б) setattr(obj, attr_name, value), в) obj.setattr(attr_name, value), г) все вышеперечисленные.

10. Укажите допустимые способы удаления атрибута объекта в Python.

- а) del obj.attr, б) obj.del(attr_name), в) obj.delattr(attr_name), г) delattr(obj, attr_name).

11. Какой атрибут используется для получения словаря, содержащего атрибуты объекта или класса?

- а) __attrs__, б) __items__, в) __dict__, г) __vars__.

12. Что будет выведено на экран в результате выполнения кода условия?

- а) {'name': 'ivan', 'age': 25}, б) {'age': 25}, в) {'name': 'ivan'}, г) ошибка.

Условие

```
class Person:
    def set_age(self, age):
        self.age = age
```

```
ivan = Person()
setattr(ivan, 'name', 'ivan')
ivan.set_age(25)
```

```
print(ivan.__dict__)
del ivan.age
```

13. Что будет выведено на экран в результате выполнения кода условия?

- а) 1 + 1, б) 2, в) ничего, г) ошибка.

Условие

```
class Test:
    x = 1
```

```
t = Test()
print(Test.x + t.x)
```

14. Что будет выведено на экран в результате выполнения кода условия?

- а) 2, б) 4, в) 6, г) ошибка.

Условие

```
class Test:
    x = 1
    def __init__(self, x):
        self.x = x
```

```
t = Test(3)
print(Test.x + t.x)
```

15. Что будет выведено на экран в результате выполнения кода условия?

- а) 2, б) 3, в) 4, г) ошибка.

Условие

```
class Test:
    x = 1
```

```
t = Test()
t.x = 2
del t.x
print(Test.x + t.x)
```

16. Какой декоратор используется для определения метода класса и позволяет работать с атрибутами класса, а не с атрибутами объекта?

- а) @staticmethod, б) @classmethod, в) @property, г) @operator.

17. Перечислите верные варианты вызова метода класса в Python.

- а) obj.method(cls), б) Class.method(cls), в) obj.method(), г) Class.method().

18. Что будет выведено на экран в результате выполнения кода условия?

- а) 0, б) 1, в) 2, г) ошибка.

Условие

```
class Test:
    @classmethod
    def m_1(cls):
        return (1 if 1>0 else 0)
```

```
obj = Test()
print(obj.m_1() + Test.m_1())
```

19. Перечислите верные варианты вызова статического метода в Python.

а) obj.method(), б) Class.method(), в) Class.method(obj), г) method().

20. Что будет выведено на экран в результате выполнения кода условия?

а) 1, б) 2, в) 3, г) ошибка.

Условие

```
class Test:
```

```
    def m_1():
```

```
        return (1 if 1>0 else 0)
```

```
    @staticmethod
```

```
    def m_2():
```

```
        return (2 if 2>0 else 0)
```

```
obj = Test()
```

```
print(obj.m_1() + Test.m_2())
```

21. Как в Python принято обозначать атрибуты класса, предназначенные для внутреннего (служебного) использования?

а) _, б) __, в) @, г) никак.

22. Что будет выведено на экран в результате выполнения кода условия?

а) 5, б) False, в) None, г) ошибка.

Условие

```
class Test:
```

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

```
        self._x = x
```

```
obj = Test(5)
```

```
print(obj._x)
```

23. Как в Python принято обозначать приватные атрибуты класса?

а) _, б) __, в) @, г) никак.

24. Что будет выведено на экран в результате выполнения кода условия?

а) 5, б) False, в) None, г) ошибка.

Условие

```
class Test:
```

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

```
        self.__x = x
```

```
obj = Test(5)
```

```
print(obj.__x)
```

25. Какая из встроенных функций предназначена для создания управляемых атрибутов класса?

а) staticmethod, б) classmethod, в) property, г) setattr.

26. Что будет выведено на экран в результате выполнения кода условия?

а) 3, б) 4, в) 5, г) ошибка.

Условие

```
class Test:
```

```
    @property
```

```
    def attr_x(self):
```

```
        return self.__x
```

```
    @attr_x.setter
```

```
    def attr_x(self, x):
```

```
        self.__x = 3
```

```
obj = Test()
```

```
obj.attr_x = 4
```

```
obj.attr_x += 1
```

```
print(obj.attr_x)
```

27. Что будет выведено на экран в результате выполнения кода условия?

- а) 7, б) Атрибут удален!, в) ничего, г) ошибка.

Условие

```
class Test:
```

```
    @property
```

```
    def attr_x(self):
```

```
        return self.__x
```

```
    @attr_x.setter
```

```
    def attr_x(self, x):
```

```
        self.__x = x
```

```
    @attr_x.deleter
```

```
    def del_attr_x(self):
```

```
        del self.__x
```

```
        print('Атрибут удален!')
```

```
obj = Test()
```

```
obj.attr_x = 7
```

```
del obj.attr_x
```

28. Укажите верные варианты вызова метода method суперкласса A из его дочернего подкласса B.

- а) super().method(), б) super(B, self).method(), в) super(A, self).method(), г) super(B, method).

29. Что будет выведено на экран в результате выполнения кода условия?

- а) 3, б) 4, в) 5, г) ошибка.

Условие

```
class A:
```

```
    def m_s(self):
```

```
        return 1
```

```
class B:
```

```
    def m_s(self):
```

```
        return 2
```

```
class C(B, A):
```

```
    def m_sub(self):
```

```
        return 3
```

```
obj = C()
```

```
print(obj.m_s() + obj.m_sub())
```

30. Что будет выведено на экран в результате выполнения кода условия?

- а) 2, б) 3, в) 4, г) ошибка.

Условие

```
class A:
```

```
    def set_1(self):
```

```
        self.__x = 1
```

```
        return self.__x
```

```
class B:
```

```
    def set_2(self):
```

```
        self.__x = 2
```

```
        return self.__x
```

```
class C(A, B): pass
```

```
obj = C()
```

```
print(obj.set_1() + obj.set_2())
```

31. Имена магических методов в Питоне выделяются следующим образом:

- а) начинаются с одного символа подчеркивания,
б) начинаются с двух символов подчеркивания,

- в) начинаются и заканчиваются одним символом подчеркивания,
г) начинаются и заканчиваются двумя символами подчеркивания.

32. Разрешается ли наследование магических методов, а также их переопределение в подклассах?

- а) не разрешается, б) разрешается, в) разрешается только наследование,
г) разрешается только переопределение.

33. Какой магический метод отвечает за операцию сложения?

- а) `__sum__`, б) `__concat__`, в) `__add__`, г) `__plus__`.

34. Что будет выведено на экран в результате выполнения кода условия?

- а) 3, б) 4, в) None, г) ошибка.

Условие

```
class N:
```

```
    def __init__(self, n):  
        self._n = n
```

```
    def __mul__(self, x):  
        return self._n + x
```

```
print(N(1)*3)
```

35. Какой базовый класс модуля `abc` стандартной библиотеки может быть использован для создания абстрактных классов в Python?

- а) `abc`, б) `__abc__`, в) `Abc`, г) `ABC`.

36. Укажите верный декоратор модуля `abc` стандартной библиотеки, который может быть использован для создания абстрактных методов?

- а) `@abstractmethod`, б) `@abstract_method`, в) `@abstract`, г) `@ABSTRACT`.

37. Что будет выведено на экран в результате выполнения кода условия?

- а) Питоша, б) `False`, в) `None`, г) ошибка.

Условие

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

```
class Test(ABC):
```

```
    @abstractmethod
```

```
    def set_name(self, name):  
        self.name = name
```

```
obj = Test()
```

```
obj.set_name('Питоша')
```

```
print(obj.name)
```

38. Какой магический метод используется для определения деструктора?

- а) `__del__`, б) `__delete__`, в) `__destroy__`, г) `__remove__`.

Критерии оценивания:

34-38 верных ответов – «отлично»;

28-34 – «хорошо»;

20-27 – «удовлетворительно».

Доцент _____ Шестопа О.В.
(подпись) (ФИО)

« ____ » _____ 2024 г.