

Лабораторная работа №4

Тема: «Объектно-ориентированное программирование на языке Python. Инкапсуляция данных, наследование, полиморфизм. Создание экземпляров объектов».

Требования к выполнению лабораторной работы №4

1. Изучите теоретическую часть к четвёртой лабораторной работе:
 - a. Теоретическая часть к четвёртой лабораторной работе (notebook_4.ipynb).
 - b. Лекция №4.
2. Создайте новый проект.
3. Запустите примеры из лабораторной работы.
4. Выполните задание согласно вашему варианту:
 - a. Вычислите свой вариант (*согласно формуле ниже*).
Если сделали не свой вариант => работа не засчитывается.
 - b. Отправьте выполненное задание в ОРИОКС (*раздел Домашние задания*).

Формат защиты лабораторных работ:

1. Продемонстрируйте выполненные задания.
2. Ответьте на вопросы по вашему коду.
3. При необходимости выполните дополнительное (*дополнительные*) задание от преподавателя.
4. Ответьте (*устно*) преподавателю на контрольные вопросы.

Список вопросов

1. Что такое наследование?
2. Что такое композиция?
3. Аннотация свойств.
4. Ромбовидное наследование.
5. Проблемы `super()`.
6. Для чего нужен метод `__str__()`?
7. Декораторы функций.

Задания

Общее задание

1. Необходимо проверять корректность вводимых данных и выводить соответствующие сообщения об ошибках.
2. Реализовать в классе методы:
 - а. конструктор;
 - б. функцию формирования строки информации об объекте.
3. Создать проект для демонстрации работы: сформировать объекты со значениями-константами и с введенными с клавиатуры значениями полей объекта. В основной ветке программы создайте три объекта класса. Вывести результаты работы на экран.
4. На основании предложенной предметной области спроектировать несколько классов, используя механизм наследования. Для каждого класса использовать отдельный модуль.
5. Предусмотреть у класса наличие полей, методов и свойств. Названия членов класса должны быть осмысленны и снабжены комментариями.
6. Один из наследников должен перегружать метод родителя.
7. Один из классов должен содержать метод, который переопределяется в одном наследнике и не переопределяется в другом.
8. Продемонстрировать работу всех объявленных методов.
9. Продемонстрировать вызов конструктора родительского класса при наследовании.

| |
|---|
| № Варианта = номер_студенческого % 2 + 1 |
|---|

Вариант №1

Класс-родитель: «Товар»

Функция-метод №1: метод должен принимать размер скидки в процентах и возвращать итоговую цену (с точностью до двух знаков после запятой).
Минимальная возможная цена = 0.01 руб.

Функция-метод №2: метод должен принимать габариты транспортировочной коробки (*ширина, высота, глубина*) и возвращать целое число, равное количеству данного товара, которое может поместиться в данную коробку (*целое число*).

Перегрузка оператора:

перезгрузить оператор «+» (*логику работы предусмотреть самостоятельно*)

Вариант №2

Класс-родитель: *выберите самостоятельно.*

Напишите декоратор класса @remember, который должен хранить данные о созданных объектах, сохраняя информацию о них в словаре. Также данный декоратор должен выдавать ошибку/исключение при создании нового объекта класса с теми же начальными аргументами, что и ранее созданный объект.

Перегрузка оператора:

перезгрузить оператор «*» (*логику работы предусмотреть самостоятельно*)