Покинтелица Тарас ИС-27

отчет по практической 16

Тема: составление программ с использованием ООП.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с ООП в IDE PyCharm Community.

Постановка задачи №1: Создайте класс «Круг», который имеет атрибут радиуса и методы для вычисления площади, длины окружности и диаметра.
Для задачи из блока 1 создать две функции, save_def и load_def, которые позволяют сохранять информацию из экземпляров класса (3 шт.) в файл и загружать ее обратно. Использовать модуль pickle для сериализации и десериализации объектов Python в бинарном формате.

Текст программы:

```
# Создайте класс «Круг», который имеет атрибут радиуса и методы
для вычисления площади,
# длины окружности и диаметра.

class Circle:
    def __init__(self,r):
        self.r = r
    def area(self):
        return self.r**2*3.14
    def len(self):
        return self.r*2*3.14
    def diameter(self):
        return self.r*2
c = Circle(200)
print(c.area())
```

Протокол программы:

125600.0

Постановка задачи №2: Создание базового класса "Транспортное средство" и его наследование для создания классов "Автомобиль" и "Мотоцикл". В классе "Транспортное средство" будут общие свойства, такие как максимальная скорость и количество колес, а классы наследники будут иметь свои уникальные свойства и методы.

Текст программы:

```python
# Создание базового класса "Транспортное средство" и его
наследование для создания классов "Автомобиль" и "Мотоцикл".
# В классе "Транспортное средство" будут общие свойства,
# такие как максимальная скорость и количество колес, а классы
наследники будут иметь свои уникальные свойства и методы.

class Transport:
    def __init__(self, max_speed, wheels):
        self.max_speed = max_speed
        self.wheels = wheels

    def move(self):
        print("Moving at a speed of", self.max_speed)

class Car(Transport):
    def __init__(self, max_speed, wheels, brand, color):
        super().__init__(max_speed, wheels)
        self.brand = brand
        self.color = color

    def honk(self):
        print("Beep beep!")

class Motorcycle(Transport):
    def __init__(self, max_speed, wheels, brand, type):
        super().__init__(max_speed, wheels)
        self.brand = brand
        self.type = type

    def wheelie(self):
        print("Performing a wheelie!")

# Создаем объекты для классов "Автомобиль" и "Мотоцикл"
car = Car(200, 4, "Tesla", "red")
motorcycle = Motorcycle(150, 2, "Harley Davidson", "cruiser")

# Вызываем методы и обращаемся к свойствам объектов
car.move()
car.honk()
print(f"Car brand: {car.brand}, color: {car.color}")
print(f"Car max speed: {car.max_speed}, number of wheels:
{car.wheels}")

motorcycle.move()
motorcycle.wheelie()
print(f"Motorcycle brand: {motorcycle.brand}, type:
{motorcycle.type}")
```

```
print(f"Motorcycle max speed: {motorcycle.max_speed}, number of
wheels: {motorcycle.wheels}")
```

Протокол программы:

Moving at a speed of 200
Beep beep!
Car brand: Tesla, color: red
Car max speed: 200, number of wheels: 4
Moving at a speed of 150
Performing a wheelie!
Motorcycle brand: Harley Davidson, type: cruiser
Motorcycle max speed: 150, number of wheels: 2

Постановка задачи №1: Для задачи из блока 1 создать две функции, save_def и load_def, которые позволяют сохранять информацию из экземпляров класса (3 шт.) в файл и загружать ее обратно. Использовать модуль pickle для сериализации и десериализации объектов Python в бинарном формате.

Текст программы:

```
# Для задачи из блока 1 создать две функции, save_def и load_def,
# которые позволяютсохранять информацию из экземпляров класса (3
шт.) в файл и загружать ее обратно.
# Использовать модуль pickle для сериализации и десериализации
объектов Python в бинарном формате.

import pickle
from PZ_16_1 import Circle


def save_def(circle, filename):
    with open(f'{filename}.bin', 'wb') as file:
        pickle.dump(circle, file)

def load_def(filename):
    with open(f'{filename}.bin', 'rb') as file:
        return pickle.load(file)


cir1 = Circle(15)
cir2 = Circle(90)
cir3 = Circle(20)
save_def(cir1,'cir1')
save_def(cir2,'cir2')
save_def(cir3,'cir3')
c = load_def('cir2')
```

Протокол программы:

Connected to pydev debugger (build 231.9225.15)
125600.0

Вывод: закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составление программ с ООП в IDE PyCharm Community.