

Практическое занятие №16

Тема: составление программ с использованием ООП в IDE PyCharm Community

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

Задание 1:

Постановка задачи:

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

Тип алгоритма: линейный

Текст программы:

```
class Circle:
    def __init__(self, radius):
        self.radius = radius

    def get_square(self):
        print("Площадь круга:", 3.1428 * int(self.radius) *
int(self.radius))

    def get_diameter(self):
        print("Диаметр круга:", self.radius * 2)

    def get_length(self):
        print("Длина круга:", 2 * 3.1428 * self.radius)

circle1 = Circle(5)

circle1.get_square(),
circle1.get_diameter()
circle1.get_length()
```

Протокол работы программы:

Площадь круга: 78.57

Диаметр круга: 10

Длина круга: 31.427999999999997

Задание 2:

Постановка задачи:

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.

Тип алгоритма: линейный

Текст программы:

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

class Car(Vehicle):
    def __init__(self, max_speed, wheels, body):
        super().__init__(max_speed, wheels)
        self.body = body

class Bike(Vehicle):
    def __init__(self, max_speed, wheels, bike_type):
        super().__init__(max_speed, wheels)
        self.bike_type = bike_type

car = Car(200, 4, 'седан')
print("Максимальная скорость:", car.max_speed, "\nкол-во колес:",
      car.wheels, "\nкузов:", car.body)
bike = Bike(300, 2, 'мотокросс')
print("Максимальная скорость:", bike.max_speed, "\nкол-во колес:",
      bike.wheels, "\nтип:", bike.bike_type)
```

Протокол работы программы:

Максимальная скорость: 200

кол-во колес: 4

кузов: седан

Максимальная скорость: 300

кол-во колес: 2

тип: мотокросс

Задание 3:

Постановка задачи:

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.

Тип алгоритма: линейный

Текст программы:

```
import pickle

class Circle:
    def __init__(self, radius):
        self.radius = radius

    def get_square(self):
        print("Площадь круга:", 3.1428 * int(self.radius) *
int(self.radius))

    def get_diameter(self):
        print("Диаметр круга:", self.radius * 2)
```

```
def get_length(self):  
    print("Длина круга:", 2 * 3.1428 * self.radius)  
  
def save_def(element, name):  
    f = open(name, 'wb')  
    pickle.dump(element, f)  
    f.close()  
  
def load_def(name):  
    f = open(name, 'rb')  
    element = pickle.load(f)  
    print(element)  
  
circle1 = Circle(5)  
circle2 = Circle(2)  
circle3 = Circle(10)  
  
save_def(circle1, '1.bin')  
load_def('1.bin')  
save_def(circle2, '2.bin')  
load_def('2.bin')  
save_def(circle3, '3.bin')  
load_def('3.bin')
```

Протокол работы программы:

```
<__main__.Circle object at 0x0000022B3D34EA50>  
<__main__.Circle object at 0x0000022B3D34EB40>  
<__main__.Circle object at 0x0000022B3D362A20>
```

Вывод: в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ с использованием ООП в IDE PyCharm Community.