

Практическое занятие № 16

Тема: практического занятия: составление программ с использованием ООП.

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

№1

Постановка задачи.

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

Текст программы:

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

class Circle():
    """
    A simple class for creating circle and counting sqware of cirlce,
    cirmunference and diametr.
    """

    def __init__(self, radius: int):
        self.radius = radius

    def square(self,):
        return round(3.14159 * self.radius ** 2, 2)

    def circumference(self):
        return round(3.14159 * 2 * self.radius, 2)
```

```

    def diameter(self):
        return 2 * self.radius

circle1 = Circle(3)
circle2 = Circle(5)

# square of circles
s1_s = circle1.square()
s2_s = circle2.square()

# circumference of circles
s1_c = circle1.circumference()
s2_c = circle2.circumference()

# diameter of circles
s1_d = circle1.diameter()
s2_d = circle2.diameter()

if __name__ == '__main__':
    # some outputs
    print(f'Площадь первого круга: {s1_s}\nПлощадь второго круга: {s2_s}\n{'-'*40}')

    print(f'Длина окружности первого круга: {s1_c}\nДлина окружности второго круга: {s2_c}\n{'-'*40}')

    print(f'Диаметр первого круга: {s1_d}\nДиаметр второго круга: {s2_d}')

```

Протокол работы программы:

Площадь первого круга: 28.27

Площадь второго круга: 78.54

Длина окружности первого круга: 18.85

Длина окружности второго круга: 31.42

Диаметр первого круга: 6

Диаметр второго круга: 10

Process finished with exit code 0

№2

Постановка задачи.

Создайте класс "Человек", который содержит информацию о имени, возрасте и поле. Создайте классы "Мужчина" и "Женщина", которые наследуются от класса "Человек". Каждый класс должен иметь метод, который выводит информацию о поле объекта.

Текст программы:

```
# Создайте класс "Человек", который содержит информацию о имени, возрасте и поле.
# Создайте классы "Мужчина" и "Женщина", которые наследуются от класса
# "Человек". Каждый класс должен иметь метод, который выводит информацию о
# поле объекта.

class Person():
    def __init__(self, name, age, sex):
        self.name = name
        self.age = age
        self.sex = sex

    def get_sex(self):
        return f"Gender: {self.sex}"

class Man(Person):
    def __init__(self, name, age):
        super().__init__(name, age, "Male")

class Woman(Person):
    def __init__(self, name, age):
        super().__init__(name, age, 'Female')

man = Man("John", 30)
female = Woman('Irina', 20)

print(man.get_sex())
print(female.get_sex())
```

Протокол работы программы:

Gender: Male

Gender: Female

Process finished with exit code 0

№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.

Текст программы:

```
import pickle

from PZ_16_26.PZ_16_26_1 import Circle

# Функция для сохранения информации экземпляра класса в файл
def save_def(*args):
    with open('circle_info.pkl', 'wb') as file:
        pickle.dump(args, file)

# Функция для загрузки информации экземпляра класса из файла
def load_def(file_name):
    with open(file_name, 'rb') as file:
        data = pickle.load(file)
        return data

circle1 = Circle(3)
circle2 = Circle(5)

# Сохранение информации об экземплярах класса в файл
save_def(circle1, circle2)
```

```
# Загрузка информации об экземплярах класса из файла
loaded_circles = load_def('circle_info.pkl')

# Вывод информации о диаметре первого круга до и после сохранения и загрузки
print(f'Диаметр первого круга до загрузки: {circle1.diameter()}')
print(f'Диаметр первого круга после загрузки:
{loaded_circles[0].diameter()} \n')

print(f'Диаметр первого круга до загрузки: {circle2.diameter()}')
print(f'Диаметр первого круга после загрузки:
{loaded_circles[1].diameter()}')
```

Протокол работы программы:

Диаметр первого круга до загрузки: 6

Диаметр первого круга после загрузки: 6

Диаметр первого круга до загрузки: 10

Диаметр первого круга после загрузки: 10

Process finished with exit code 0

Вывод:

В ходе практического занятия №16 по использованию ООП в программировании на языке Python и с использованием IDE PyCharm Community, были изучены и применены следующие основные концепции:

1. Программно-ориентированное программирование (ООП): Мы создали классы, такие как Circle, Person, Man, и Woman, используя понятия атрибутов (например, радиус, имя, возраст, пол) и методов (например, вычисление площади, длины окружности и пола). Это показывает понимание структуры классов и их взаимодействия.
2. Инит-метод (constructor): В классах использовались `__init__`-методы для инициализации атрибутов объектов при их создании.
3. Методы доступа к атрибутам: Мы создали методы, такие как `square`, `circumference`, и `diameter` для обращения к атрибутам класса и их использования для вычислений.
4. Наследование: Классы Man и Woman наследовались от общего класса Person, что позволило использовать общие атрибуты и методы внутри этих специфических классов.
5. Модуль `pickle`: В рамках задачи №3, использован модуль `pickle` для сериализации и десериализации экземпляров классов Circle, позволяя сохранять и загружать информацию о них в файл.

Изученные навыки:

- Компонование и настройка классов и объектов с использованием атрибутов и методов.

- Применение наследования в классовой модели.
- Использование функций для работы с данными, сохранения и загрузки из файлов.
- Работа с модулями, в данном случае pickle, для бинарной сериализации и десериализации.

Итоговый протокол:

- В занятии были успешно созданы классы, соответствующие заданию, и выполнялись соответствующие вычисления.
- Мы реализовали функции для сохранения и загрузки данных о классовых объектах, что показывает знакомство с файловой системой инициализацией объектов.
- Результаты работы программы, как и ожидаемо, были получены корректно.

Конечно, в процессе обучения были приобретены навыки в области создания и управления классами с использованием объектно-ориентированного подхода на примере Python и PyCharm.

Готовые программные коды выложены на GitHub.