

Практическое занятие № 16

Тема: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ с использованием ООП в IDE PyCharm Community.

Постановка задачи. Создайте класс "Машина" с атрибутами "марка", "модель" и "год выпуска". Напишите метод, который выводит информацию о машине в формате "Марка: марка, Модель: модель, Год выпуска: год".

Тип алгоритма:циклический

Текст программы:

```
class Car:
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year

    def display_info(self):
        print(f"Марка: {self.make}, Модель: {self.model}, Год выпуска: {self.year}")

car1 = Car("Toyota", "Camry", 2020)
car2 = Car("Ford", "Mustang", 1969)
car1.display_info()
car2.display_info()

#Создайте базовый класс "Форма" со свойствами "цвет" и "тип". От этого класса унаследуйте класс "Круг" и
#добавьте в него свойство "радиус". Определите методы вычисления площади и периметра.
import math

class Shape:
```

```
def __init__(self, color, shape_type):
    self.color = color
    self.shape_type = shape_type

def display_info(self):
    print(f"Цвет: {self.color}, Тип: {self.shape_type}")

class Circle(Shape):
    def __init__(self, color, radius):
        super().__init__(color, "Круг")
        self.radius = radius

    def calculate_area(self):
        return math.pi * (self.radius ** 2)

    def calculate_perimeter(self):
        return 2 * math.pi * self.radius

    def display_info(self):
        super().display_info()
        print(f"Радиус: {self.radius}, Площадь: {self.calculate_area()}, Периметр: {self.calculate_perimeter()}")

circle = Circle("Красный", 5)
circle.display_info()

#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
import pickle

class Car:
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year

    def display_info(self):
        print(f"Марка: {self.make}, Модель: {self.model}, Год выпуска: {self.year}")
```

```
def save_def(cars, filename):
    try:
        with open(filename, 'wb') as file:
            pickle.dump(cars, file)
            print("Информация успешно сохранена в файл.")
    except Exception as e:
        print(f"Ошибка при сохранении данных: {e}")

def load_def(filename):
    try:
        with open(filename, 'rb') as file:
            cars = pickle.load(file)
            print("Информация успешно загружена из файла.")
            return cars
    except Exception as e:
        print(f"Ошибка при загрузке данных: {e}")
        return []

car1 = Car("Toyota", "Camry", 2020)
car2 = Car("Ford", "Mustang", 1969)
car3 = Car("Honda", "Civic", 2018)
cars = [car1, car2, car3]
save_def(cars, 'cars.pkl')
loaded_cars = load_def('cars.pkl')
for car in loaded_cars:
    car.display_info()
```

Протокол работы программы:

Марка: Toyota, Модель: Camry, Год выпуска: 2020

Марка: Ford, Модель: Mustang, Год выпуска: 1969

Вывод: В процессе выполнения закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с использованием ООП в IDE PyCharm

Community. Были использованы языковые конструкции `class`, `def`

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub