

Практическое занятие №16

Тема: Составление программ с использованием ООП в IDE PyCharm Community.

Цели: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с 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".

Текст программы:

```
# 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".

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: {self.year}")

car = Car("Toyota", "Corolla", 2015)

if __name__ == "__main__":
    car.display_info()
```

Протокол Программы:

Марка: Toyota, Модель: Corolla, Год: 2015

Постановка задачи:

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.

Текст программы:

```
# 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

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

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

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

if __name__ == "__main__":

    circle = Circle("Красный", 5)

    area = circle.area()
    perimeter = circle.perimeter()

    print(f"Площадь: {area:.2f}")
    print(f"Периметр: {perimeter:.2f}")
```

Протокол Программы:

Площадь: 78.54

Периметр: 31.42

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:

```
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

from PZ_16_1 import Car # type: ignore

def save_def(filename, objects):
    with open(filename, 'wb') as file:
        pickle.dump(objects, file)

def load_def(filename):
    with open(filename, 'rb') as file:
        objects = pickle.load(file)
    return objects

car1 = Car("Toyota", "Corolla", 2015)
car2 = Car("Honda", "Civic", 2018)
car3 = Car("Ford", "Mustang", 2020)

objects = [car1, car2, car3]
save_def("cars.pkl", objects)

loaded_objects = load_def("cars.pkl")

for car in loaded_objects:
    car.display_info()
```

Марка: Toyota, Модель: Corolla, Год: 2015

Марка: Honda, Модель: Civic, Год: 2018

Марка: Ford, Модель: Mustang, Год: 2020

Вывод:

Я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навык и составление программ с ООП в IDE PyCharm Community.