

Практическое занятие №16

Тема: 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.

Постановка задачи №1:

Создайте класс «Банк», который имеет атрибуты суммы денег и процентной ставки.

Добавьте методы для вычисления процентных начислений и снятия денег

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: циклический

Текст программы:

```
import pickle

class Bank:
    def __init__(self, money, interest_rate):
        self.money = money
        self.interest_rate = interest_rate

    def calculate_interest(self):
        return self.money * self.interest_rate / 100

    def withdraw_money(self, amount):
        if amount <= self.money:
            self.money -= amount
        else:
            print("Not enough money in the account.")

def save_def(objects, filename):
    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

bank1 = Bank(1000, 2)
```

```
bank2 = Bank(5000, 3)
bank3 = Bank(200, 1.5)

banks_to_save = [bank1, bank2, bank3]
save_def(banks_to_save, 'banks.pkl')

loaded_banks = load_def('banks.pkl')

for bank in loaded_banks:
    print("Initial amount in the account:", bank.money)
    print("Interest for the account:", bank.calculate_interest())
    bank.withdraw_money(100)
    print("Amount in the account after withdrawal:", bank.money)
    print()
```

Протокол программы:

Initial amount in the account: 1000

Interest for the account: 20.0

Amount in the account after withdrawal: 900

Initial amount in the account: 5000

Interest for the account: 150.0

Amount in the account after withdrawal: 4900

Initial amount in the account: 200

Interest for the account: 3.0

Amount in the account after withdrawal: 100

Process finished with exit code 0

Постановка задачи №2:

Создайте класс "Фрукт", который содержит информацию о наименовании и весе фрукта. Создайте классы "Яблоко" и "Апельсин", которые наследуются от класса "Фрукт" и содержат информацию о цвете.

Тип алгоритма: циклический

Текст программы:

```
class Fruit:
    def __init__(self, name, weight):
        self.name = name
        self.weight = weight

class Apple(Fruit):
    def __init__(self, name, weight, color):
        super().__init__(name, weight)
        self.color = color

class Orange(Fruit):
    def __init__(self, name, weight, color):
        super().__init__(name, weight)
        self.color = color

apple = Apple("Яблоко", 150, "красное")
orange = Orange("Апельсин", 200, "оранжевый")

print(f"{apple.name} весит {apple.weight} грамм, цвета {apple.color}")
print(f"{orange.name} весит {orange.weight} грамм, цвета {orange.color}")
```

Протокол программы:

Яблоко весит 150 грамм, цвета красное

Апельсин весит 200 грамм, цвета оранжевый

Process finished with exit code 0

Вывод: в процессе выполнения практического задания я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ с применением множеств в IDE PyCharm Community.

