

Практическое занятие №16-(1, 2, 3)

Тема: Составление программ с использованием ООП в IDE PyCharm.

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

Постановка задачи:

16.1 Создайте класс «Банк», который имеет атрибуты суммы денег и процентной ставки. Добавьте методы для вычисления процентных начислений и снятия денег.

16.2 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.

16.3 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 программ:

16.1

```
class Bank:
    def __init__(self, amount, interest_rate):
        self.amount = amount # сумма денег на счету
        self.interest_rate = interest_rate # годовая процентная ставка

    def calculate_interest(self):
        """Метод для вычисления процентных начислений"""
        interest = self.amount * (self.interest_rate / 100)
        return interest

    def withdraw(self, amount_to_withdraw):
        """Метод для снятия денег"""
        if amount_to_withdraw > self.amount:
            return "Недостаточно средств на счету."
        self.amount -= amount_to_withdraw
        return f"Вы сняли {amount_to_withdraw}. Остаток на счету: {self.amount}"

# Пример использования
bank_account = Bank(10000, 5.0)
print(f"Процентные начисления: {bank_account.calculate_interest()}")
print(bank_account.withdraw(2000))
```

16.2

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

    def __str__(self):
        return f"Фрукт: {self.name}, Вес: {self.weight} кг"

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

    def __str__(self):
        return f"Фрукт: {self.name}, Вес: {self.weight} кг, Цвет: {self.color}"

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

    def __str__(self):
        return f"Фрукт: {self.name}, Вес: {self.weight} кг, Цвет: {self.color}"

# Пример использования
apple = Apple("Яблоко", 0.2, "Зеленый")
orange = Orange("Апельсин", 0.3, "Оранжевый")

print(apple)
print(orange)
```

16.3

```

import pickle

# Класс Bank для работы с банковским счетом
class Bank:
    def __init__(self, amount, interest_rate):
        self.amount = amount # сумма денег на счету
        self.interest_rate = interest_rate # годовая процентная ставка

    def calculate_interest(self):
        """Метод для вычисления процентных начислений"""
        interest = self.amount * (self.interest_rate / 100)
        return interest

    def withdraw(self, amount_to_withdraw):
        """Метод для снятия денег"""
        if amount_to_withdraw > self.amount:
            return "Недостаточно средств на счету."
        self.amount -= amount_to_withdraw
        return f"Вы сняли {amount_to_withdraw}. Остаток на счету: {self.amount}"

# Функция для сохранения объектов в файл
def save_def(objects, filename):
    with open(filename, 'wb') as file:
        pickle.dump(objects, file)
    print(f"Данные сохранены в файл {filename}")

# Функция для загрузки объектов из файла
def load_def(filename):
    with open(filename, 'rb') as file:
        objects = pickle.load(file)
    print(f"Данные загружены из файла {filename}")
    return objects

# Пример использования для класса Bank
bank1 = Bank(10000, 5.0)
bank2 = Bank(5000, 3.0)
bank3 = Bank(2000, 7.0)

# Сохранение объектов
banks = [bank1, bank2, bank3]
save_def(banks, 'bank_data.pkl')

# Загрузка объектов
loaded_banks = load_def('bank_data.pkl')
for bank in loaded_banks:
    print(f"Сумма: {bank.amount}, Процентная ставка: {bank.interest_rate}")

```

Протокол работы программы:

16.1

Процентные начисления: 500.0

Вы сняли 2000. Остаток на счету: 8000

16.2

Фрукт: Яблоко, Вес: 0.2 кг, Цвет: Зеленый

Фрукт: Апельсин, Вес: 0.3 кг, Цвет: Оранжевый

16.3

Данные сохранены в файл bank_data.pkl

Данные загружены из файла bank_data.pkl

Сумма: 10000, Процентная ставка: 5.0

Сумма: 5000, Процентная ставка: 3.0

Сумма: 2000, Процентная ставка: 7.0

Вывод: В процессе выполнения практического задания выработала навыки составления программ с ООП в IDE PyCharm. Выполнены разработка кода, отладка, тестирование, оптимизация. Готовые программные коды выложены на GitHub.