Отчёт Практическая работа №16
Неклюдов Витлаий ИС-27

Тема: Наименование практического занятия: составление программ с использованием ООП

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

Задание 1
Постановка задачи: Создайте класс «Банк», который имеет атрибуты суммы денег и процентной ставки.

Код программы:

```
class Bank:
    def __init__(self, amount, percent):
        self.amount = amount
        self.percent = percent

    def calculate_interest(self):
        interest = self.amount * (self.percent / 100)
        return interest

    def withdraw(self, amount_to_withdraw):
        if amount_to_withdraw <= self.amount:
            self.amount -= amount_to_withdraw
            return f"{amount_to_withdraw}$ снято."
        else:
            return "недостаточно средств"


safe_1 = Bank(5000, 10)
safe_2 = Bank(10000, 15)
```

Протокол работы:
Process finished with exit code 0

Задание 2
Постановка задачи: Создайте класс "Животное", который содержит информацию о виде и возрасте животного. Создайте классы "Собака" и "Кошка", которые наследуются от класса "Животное" и содержат информацию о породе.

Код программы:

```
class Animal:
    def __init__(self, kind, age):
        self.kind = kind
        self.age = age
```

```
class Dog(Animal):
    def __init__(self, kind, age, breed):
        super().__init__(kind, age)
        self.breed = breed


class Cat(Animal):
    def __init__(self, kind, age, breed):
        super().__init__(kind, age)
        self.breed = breed


dog = Dog("Собака", 5, "Лабрадор")
cat = Cat("Кошка", 3, "Персидская")


print(f"Собака: Вид - {dog.kind}, Возраст - {dog.age}, Порода -
{dog.breed}")
print(f"Кошка: Вид - {cat.kind}, Возраст - {cat.age}, Порода -
{cat.breed}")
```

Протокол работы: Process finished with exit code 0

Задание 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

Код программы:

```
def save_def(safe):
    with open("safes", 'wb') as file:
        pickle.dump(safe, file)


def load_def():
    with open("safes", 'rb') as file:
        return pickle.load(file)


save_def([safe_1, safe_2])

loaded_safes = load_def()

for safe in loaded_safes:
    print(f"Сумма: {safe.amount}")
    print(f"Процент: {safe.percent}")
    print(f"Прибыль год: {safe.calculate_interest()}")
```

Протокол работы:
Собака: Вид - Собака, Возраст - 5, Порода - Лабрадор
Кошка: Вид - Кошка, Возраст - 3, Порода - Персидская
Сумма: 5000
Процент: 10
Прибыль год: 500.0
Сумма: 10000
Процент: 15
Прибыль год: 1500.0

Process finished with exit code 0

Вывод: выполняя практическую работу, я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрёл навыки составления программ с ООП в IDE PyCharm Community.