

Кафедра компьютерной инженерии и моделирования

Порозов Кирилл Сергеевич

отчет по практической работе №8
по дисциплине «**ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ**»

Направление подготовки:
09.03.04 "Программная инженерия"



Оценка -

Симферополь, 2022

Практическая работа №8. Тема: "Объектно-ориентированное программирование"

Цель работы: Изучить парадигму объектно-ориентированного программирования, преимущества и недостатки парадигмы, научиться создавать простейшие приложения на языке высокого уровня, реализующие ООП подход к созданию программ.

Описание ключевых понятий:

(при необходимости)

Перед выполнением практической работы изучена следующая литература:

1. Презентация лектора курса: «Объектно-ориентированное программирование»
2. Прослушана видеолекция и посещены практические занятия Милюкова Виктора Васильевича.
3. Прочитаны материалы статей:

Написана программа на языке Python, демонстрирующая реализацию ООП стиля:

Фото кода:

```

class Animal(object):
    def __init__(self, name: str, sex: str, age: int, alive: bool):
        self.__age = age
        self.__alive = alive
        self._sex = sex
        self._name = name

    @property
    def is_alive(self):
        return self.__alive

    def greeting(self):
        return f"Hello, my name is {self._name}"

    def get_name(self):
        return self._name

    def get_sex(self):
        return self._sex

    def get_age(self):
        return self.__age

    def set_name(self, new_name: str):
        self._name = new_name
        return

class Human(Animal):
    def __init__(self, name: str, sex: str, age: int, alive: bool, birthday: str, nationality: str, country: str, profession: str, is_working: bool = False):
        super().__init__(name, sex, age, alive)
        self.is_working = is_working
        self.profession = profession
        self.country = country
        self.nationality = nationality
        self.birthday = birthday

    @property
    def go_to_work(self):
        if self.is_alive:
            @property
            def go_to_work(self):
                if self.is_alive:
                    self.is_working = True
                    return

    def stop_working(self):
        if self.is_alive:
            self.is_working = False

    def is_birthday(self, string: str):
        if string == self.birthday: return 1
        return 0

    def set_sex(self, new_sex: str):
        self._sex = new_sex
        return

horse = Animal('horse', 'female', 12, False)
bird = Animal('bird', 'male', 1, True)

peter = Human('Peter', 'combat helicopter', 100, True, '14/10/2000', 'Russian', 'Italia', 'software engineer', True)

print(f'{horse.get_name()} is alive' if horse.is_alive else f'{horse.get_name()} died')
print(f'{peter.get_name()} is alive' if peter.is_alive else f'{peter.get_name()} died')
print(peter.get_name(), peter.greeting(), peter.set_sex("Male"), peter.get_sex())
print('working now' if peter.is_working else 'is not working now', peter.go_to_work, 'working now' if peter.is_working else 'is not working now')
try:
    print(peter.__alive)
except AttributeError as aerr:
    print("This attribute is closed for user. ", aerr)

try:
    # сокрытие в питоне - неполное, полностью запретить доступ к полю мы не можем. Этот режим доступа - protected.
    print(peter._name)
except AttributeError as aerr:
    print("This attribute is closed for user. ", aerr)

try:
    # как мы видим - в питоне можно обойти даже приватный доступ к переменной, однако, делать так нельзя.

```

Более подробно – на странице задания на гитхабе: [JustBlood/Prog_Paradigms \(github.com\)](https://github.com/JustBlood/Prog_Paradigms)

Проект представлен преподавателю в электронной форме, продемонстрирована работоспособность программы, разъяснены детали программного кода.

Вопросы, заданные преподавателем: