

Практическое занятие №16

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

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

Постановка задачи 1 блока.

1. Создайте класс "Студент", который имеет атрибуты имя, фамилия и оценки. Добавьте методы для вычисления среднего балла и определения, является ли студент отличником

Текст программы:

```
# !usr/local/bin/python
# coding: UTF-8
# Создайте класс "Студент", который имеет атрибуты имя, фамилия и
оценки. Добавьте методы для вычисления среднего балла
# и определения, является ли студент отличником
```

```
class Student:
    def __init__(self, first_name, last_name, grades):
        self.first_name = first_name
        self.last_name = last_name
        self.grades = grades

    def sr(self):
        return sum(self.grades) / len(self.grades)

    def otl(self):
        return all(grade == 5 for grade in self.grades)
```

Пример использования

```
student1 = Student("Алиса", "Мир", [5, 5, 5])
```

```
student2 = Student("Наташа", "Москов", [3, 4, 4, 4])
```

```
student3 = Student("Андрей", "Железный", [5, 5, 5, 4])
```

```
print(f'{student1.first_name} {student1.last_name} имеет средний бал  
равный {student1.sr()}')
```

```
print(f'{student2.first_name} {student2.last_name} имеет средний бал  
равный {student2.sr()}')
```

```
print(f'{student3.first_name} {student3.last_name} имеет средний бал  
равный {student3.sr()}')
```

```
print(f'\n{student1.first_name} {student1.last_name} является  
отличником: {student1.otl()}')
```

```
print(f'{student2.first_name} {student2.last_name} является  
отличником: {student2.otl()}')
```

```
print(f'{student3.first_name} {student3.last_name} является  
отличником: {student3.otl()}')
```

Протокол работы программы:

Алиса Мир имеет средний бал равный 5.0

Наташа Москов имеет средний бал равный 3.75

Андрей Железный имеет средний бал равный 4.75

Алиса Мир является отличником: True

Наташа Москов является отличником: False

Андрей Железный является отличником: False

Process finished with exit code 0

Постановка задачи 2 блока.

Создание базового класса "Животное" и его наследование для создания классов "Собака", "Кошка". В классе "Животное" будут общие методы, такие как "дышать" и "питаться", а классы наследники

будут иметь свои уникальные методы и свойства такие как "гавкать" и "мурлыкать".

Текст программы:

```
# Вариант 19. Создание базового класса "Животное" и его
наследование для создания классов "Собака", "Кошка". В классе
# "Животное" будут общие методы, такие как "дышать" и "питаться",
а классы наследники будут иметь свои уникальные
# методы и свойства такие как "гавкать" и "мурлыкать".
```

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

    def breathe(self):
        print(f'{self.name} дышит')

    def eat(self):
        print(f'{self.name} питается')

class Dog(Animal):
    def bark(self):
        print(f'{self.name} гавкает')

class Cat(Animal):
    def meow(self):
        print(f'{self.name} мурлыкает')

d = Dog('Щеночек')
c = Cat('Котенок')

d.breathe()
d.eat()
d.bark()
```

```
print()
```

```
c.breathe()
```

```
c.eat()
```

```
c.meow()
```

Протокол работы программы:

Щеночек дышит

Щеночек питается

Щеночек гавкает

Котенок дышит

Котенок питается

Котенок мурлыкает

Process finished with exit code 0

Постановка задачи 3 блока:

Для задачи из блока I 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

Текст программы:

```
# !/usr/local/bin/python
```

```
# coding: UTF-8
```

```
# Для задачи из блока I 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
```

```
import pickle
class Student:
    def __init__(self, first_name, last_name, grades):
        self.first_name = first_name
        self.last_name = last_name
        self.grades = grades

    def sr(self):
        return sum(self.grades) / len(self.grades)

    def otl(self):
        return all(grade == 5 for grade in self.grades)

# Пример использования
student1 = Student("Алиса", "Мир", [5, 5, 5])
student2 = Student("Наташа", "Москов", [3, 4, 4, 4])
student3 = Student("Андрей", "Железный", [5, 5, 5, 4])

print(f"{student1.first_name} {student1.last_name} имеет средний бал  
равный {student1.sr()}")
print(f"{student2.first_name} {student2.last_name} имеет средний бал  
равный {student2.sr()}")
print(f"{student3.first_name} {student3.last_name} имеет средний бал  
равный {student3.sr()}")

print(f"\n{student1.first_name} {student1.last_name} является  
отличником: {student1.otl()}")
print(f"{student2.first_name} {student2.last_name} является  
отличником: {student2.otl()}")
print(f"{student3.first_name} {student3.last_name} является  
отличником: {student3.otl()}")

print()
student1 = ["Алиса", "Мир", [5, 5, 5]]
student2 = ["Наташа", "Москов", [3, 4, 4, 4]]
```

```
student3 = ["Андрей", "Железный", [5, 5, 5, 4]]
```

```
def save_data(student, filename):  
    with open(filename, 'wb') as file:  
        pickle.dump(student, file)  
        print(f'Data saved to {filename}')
```

```
def load_data(filename):  
    with open(filename, 'rb') as file:  
        student = pickle.load(file)  
        return student
```

```
save_data(student1, "student1.bin")  
save_data(student2, "student2.bin")  
save_data(student3, "student3.bin")
```

```
loaded_student1 = load_data("student1.bin")  
loaded_student2 = load_data("student2.bin")  
loaded_student3 = load_data("student3.bin")
```

```
print("\nВывод из файла:", loaded_student1, loaded_student2,  
loaded_student3, sep='\n')
```

Протокол работы программы:

Алиса Мир имеет средний бал равный 5.0

Наташа Москов имеет средний бал равный 3.75

Андрей Железный имеет средний бал равный 4.75

Алиса Мир является отличником: True

Наташа Москов является отличником: False

Андрей Железный является отличником: False

Data saved to student1.bin

Data saved to student2.bin

Data saved to student3.bin

Вывод из файла:

['Алиса', 'Мир', [5, 5, 5]]

['Наташа', 'Москов', [3, 4, 4, 4]]

['Андрей', 'Железный', [5, 5, 5, 4]]

Process finished with exit code 0

Вывод: закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ с использованием ООП в IDE PyCharm Community.