

Практическое занятие № 17

Тема: Составление программ с регулярными выражениями в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с классами IDE PyCharm Community

№ 17.1

Постановка задачи.

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

Текст программы:

#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, name, surname, grades):
        self.name = name
        self.surname = surname
        self.grades = grades

    def average_grade(self):
        avg = sum(self.grades) / len(self.grades)
        return round(avg, 2)

    def is_excellent(self):
        return self.average_grade() == 5

student_1 = Student('Ivan', 'Smirnov', [4, 5, 4])
student_2 = Student('Katya', 'Ivanova', [5, 5, 4])

print(f"Средний балл студента {student_1.name} {student_1.surname}: {student_1.average_grade()}")
print(f"Студент {student_1.name} {student_1.surname} является отличником: {student_1.is_excellent()}")

print()

print(f"Средний балл студента {student_2.name} {student_2.surname}: {student_2.average_grade()}")
print(f"Студент {student_2.name} {student_2.surname} является отличником: {student_2.is_excellent()}")
```

Process finished with exit code 0

Протокол работы программы:

Средний балл студента Ivan Smirnov: 4.33

Студент Ivan Smirnov является отличником: False

Средний балл студента Katya Ivanova: 5.0

Студент Katya Ivanova является отличником: True

№ 17.2

Постановка задачи.

Создание базового класса "Животное" и его наследование для создания классов "Собака" и "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 второй программы:

#Создание базового класса "Животное" и его наследование для создания классов
#"Собака" и "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 murr(self):
        print(f"{self.name} мурлыкает")

animal = Animal("Животное")
```

```
dog = Dog("Бобик")  
cat = Cat("Мурка")
```

```
animal.breathe()  
animal.eat()
```

```
dog.breathe()  
dog.eat()  
dog.bark()
```

```
cat.breathe()  
cat.eat()  
cat.murr()
```

Протокол работы программы:

Животное дышит
Животное питается
Бобик дышит
Бобик питается
Бобик гавкает
Мурка дышит
Мурка питается
Мурка мурлыкает

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с классами в IDE PyCharm Community. Были использованы языковые конструкции class и др. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.