

Практическое занятие №17.

Тема: составление программ с использованием ООП.

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

Постановка задачи: Создайте класс «Человек» с атрибутами «имя», «возраст» и «пол». Напишите метод, который выводит информацию о человеке в формате «Имя: имя, Возраст: возраст, Пол: пол».

Тип алгоритма: Линейный.

Текст программы:

```
# Создайте класс "Человек" с атрибутами "имя", "возраст" и "пол". Напишите метод,  
# который выводит информацию о человеке в формате "Имя: имя, Возраст: возраст,  
# Пол: пол".  
  
class Human: # Создание класса Человек  
    def __init__(self, name, age, gender):  
        self.name = name  
        self.age = age  
        self.gender = gender  
  
    def __str__(self): # Переопределение магического метода при обращении к экземпляру через print  
        return f"Имя: {self.name}, Возраст: {self.age}, Пол: {self.gender}"  
  
print(Human("Алина", 20, 'женский'))
```

Протокол работы программы:

Имя: Алина, Возраст: 20, Пол: женский

Process finished with exit code 0

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 «мяукать».

Тип алгоритма: Линейный.

Текст 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: # Создание класса животное
    def __init__(self, name):
        self.name = name

    def breathe(self):
        return "дышит"

    def eat(self):
        return "кушает"

class Dog(Animal): # Создание класса собака
    def bark(self):
        return "гавкает"

class Cat(Animal): # Создание класса кошка
    def purr(self):
        return "мурлыкает"

dog = Dog('Пудель')
cat = Cat('Девон-рекс')
print(dog.breathe())
print(cat.breathe())
print(dog.eat())
print(cat.eat())
print(dog.bark())
print(cat.purr())
```

Протокол работы программы:

дышит
дышит
кушает
кушает
гавкает
мурлыкает

Process finished with exit code 0

Выполнила студентка группы ИС-21 Гудзенко Екатерина

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