

## Практическое занятие №16.

Тема: «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.»

### Постановка задачи №1:

```
# Создайте класс "Компьютер" с атрибутами "марка", "процессор" и "оперативная  
# память". Напишите метод, который выводит информацию о компьютере в формате  
# "Марка: марка, Процессор: процессор, Оперативная память: память".
```

### Тип алгоритма №1: Линейный

### Код алгоритма №1:

```
class Computer:  
    def __init__(self, brand, processor, ram):  
        self.brand = brand  
        self.processor = processor  
        self.ram = ram  
  
    def display_info(self):  
        print(f"Бренд: {self.brand}, Процессор: {self.processor}, Оперативная  
        память: {self.ram}")  
  
computer = Computer("Gigabyte", "Intel Core I5", "16 GB")  
computer.display_info()
```

### Вывод программы №1:

```
Бренд: Gigabyte, Процессор: Intel Core I5, Оперативная память: 16 GB
```

## Постановка задачи №2:

```
# Создайте класс "Человек", который содержит информацию о имени, возрасте и поле.  
# Создайте классы "Мужчина" и "Женщина", которые наследуются от класса  
# "Человек". Каждый класс должен иметь метод, который выводит информацию о  
# поле объекта.
```

## Тип алгоритма: Линейный

## Код алгоритма №2:

```
class Human:  
    def __init__(self, name, age, gender):  
        self.name = name  
        self.age = age  
        self.gender = gender  
  
class Man(Human):  
    def __init__(self, name, age):  
        super().__init__(name, age, "Male")  
  
    def display_info(self):  
        print(f"{self.name} - мужчина, ему {self.age} год(а)/лет.")  
  
class Woman(Human):  
    def __init__(self, name, age):  
        super().__init__(name, age, "Female")  
  
    def display_info(self):  
        print(f"{self.name} - девушка, ей {self.age} год(а)/лет.")  
  
man = Man("Антон", 21)  
  
woman = Woman("Нина", 23)  
  
man.display_info()  
  
woman.display_info()
```

## Вывод алгоритма №2:

```
Антон - мужчина, ему 21 год(а)/лет.  
Нина - девушка, ей 23 год(а)/лет.
```

## Постановка задачи №3:

```
# 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.
```

## Тип алгоритма №3: Линейный

## Код алгоритма №3:

```
import pickle

class Computer:
    def __init__(self, brand, processor, ram):
        self.brand = brand
        self.processor = processor
        self.ram = ram

    def display_info(self):
        print(f"Брэнд: {self.brand}, Процессор: {self.processor}, Оперативная
память: {self.ram}")

def save_def(obj, filename):
    with open(filename, "wb") as f:
        pickle.dump(obj, f)

def load_def(filename):
    with open(filename, "rb") as f:
        obj = pickle.load(f)
    return obj
```

## Вывод алгоритма №3:

```
Брэнд: Apple, Процессор: M1, Оперативная память: 16GB
Брэнд: Dell, Процессор: Intel i7, Оперативная память: 32GB
Брэнд: HP, Процессор: AMD Ryzen 9, Оперативная память: 64GB
```

**Вывод:** «Я закрепил понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с ООП в IDE PyCharm Community.»