

## Практическое занятие № 16

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

**Цель:** Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием ООП в IDE PyCharm Community

### Постановка задачи.

1. Создать класс Животное с атрибутами "имя" и "вид", написать метод который выводит информацию о животном в формате "Имя: имя, Вид: вид"

### Текст программы:

```
#Создать класс Животное с атрибутами "имя" и "вид"
#Написать метод который выводит информацию о животном
#в формате "Имя: имя, Вид: вид"

#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.

import pickle

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

    def info(self):
        print(f"Имя: {self.name}, Вид: {self.species}")

def save(objects, filename):
    with open(filename, 'wb') as f:
        pickle.dump(objects, f)

def load(filename):
    with open(filename, 'rb') as f:
        return pickle.load(f)

animal1 = Animal("Барсик", "Собака")
animal2 = Animal("Мурзик", "Кот")
animal3 = Animal("Пушок", "Хомяк")

animals = [animal1, animal2, animal3]

save(animals, 'animals.pkl')
loaded_animals = load('animals.pkl')

for animal in loaded_animals:
    animal.info()
```

### Постановка задачи.

2. Создать базовый класс "Человек" со свойствами "Имя", "Возраст" и "Пол". От этого

класса унаследовать классы "Мужчина" и "Женщина", добавить свойства связанные с соц положением

### Текст программы:

```
#Создать базовый класс "Человек" со свойствами "Имя", "Возраст" и "Пол".
#От этого класса унаследовать классы "Мужчина" и "Женщина"
#Добавить свойства связанные с соц положением

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, job_title, salary):
        super().__init__(name, age, "мужчина")
        self.job_title = job_title
        self.salary = salary

class Woman(Human):
    def __init__(self, name, age, marital_status, children):
        super().__init__(name, age, "женщина")
        self.marital_status = marital_status
        self.children = children

man = Man("Петя", 30, "Программист", 80000)
woman = Woman("Маша", 28, "Замужем", 2)

print(man.name)
print(man.salary)
print(woman.name)
print(woman.children)
```

Петя  
80000  
Маша  
2

Вывод: Были закреплены усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобретены навыки составление программ с использованием ООП в IDE PyCharm Community