

### **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:**

#### **Задача №1**

Создайте класс «Календарь», который имеет атрибуты год, месяц и день. Добавьте методы для определения дня недели, проверки на високосный год и определения количества дней в месяце.

#### **Задача №2**

Создайте базовый класс "Животное" со свойствами "вид", "количество лап", "цвет шерсти". От этого класса унаследуйте класс "Собака" и добавьте в него свойства "кличка" и "порода".

**Текст программы**

#### **Задача №1**

#Создайте класс «Календарь», который имеет атрибуты год, месяц и день. Добавьте  
#методы для определения дня недели, проверки на високосный год и определения  
#количества дней в месяце..

```
import datetime
```

```
class Calendar:
```

```
    def __init__(self, year, month, day):  
        self.year = year  
        self.month = month  
        self.day = day
```

```
    def leap_year(self):  
        if self.year % 4 == 0:  
            if self.year % 100 == 0:  
                if self.year % 400 == 0:  
                    return True  
                else:  
                    return False  
            else:  
                return True  
        else:  
            return False
```

```
    def days_in_a_month(self):  
        if self.month in [1, 3, 5, 7, 8, 10, 12]:  
            return 31  
        elif self.month in [4, 6, 9, 11]:  
            return 30
```

```
    else:
        if self.leap_year():
            return 29
        else:
            return 28

    def day_of_week(self):
        days = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота',
'Воскресенье']
        return days[datetime.date(self.year, self.month, self.day).weekday()]

#Тест
c = Calendar(2023, 5, 23)
print('Високосный год? ', c.leap_year())
print('Дней в месяце: ', c.days_in_a_month())
print('День недели: ', c.day_of_week())
```

## Задача №2

#Создайте базовый класс "Животное" со свойствами "вид", "количество лап", "цвет  
#шерсти". От этого класса унаследуйте класс "Собака" и добавьте в него свойства  
#"кличка" и "порода"..

```
class Animal:
    def __init__(self, species, legs, color):
        self.species = species
        self.legs = legs
        self.color = color

class Dog(Animal):
    def __init__(self, species, legs, color, name, breed):
        Animal.__init__(self, species, legs, color)
        self.name = name
        self.breed = breed
```

```
#Тест
d = Dog('Собака', 4, 'Коричневый', 'Шарик', 'Лабрадор')
print('Вид: ', d.species)
print('Количество лап: ', d.legs)
print('Цвет: ', d.color)
print('Кличка: ', d.name)
print('Порода: ', d.breed)

c = Animal('Птица', 2, 'Розовый')
print('Вид: ', c.species)
```

```
print('Количество лап: ', c.legs)  
print('Цвет: ', c.color)
```

**Вывод:**

В процессе выполнения практического занятия я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ с ООП в IDE PyCharm Community.