

## Практическая работа № 16

**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.

### Задача Блока 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.

### Код программы:

```
"""
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 re
from datetime import datetime

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

    def __str__(self):
        return (f"\nГод: {self.year}\n"
                f"Месяц: {self.month:02d}\n"
                f"Число: {self.day:02d}\n"
                )

    def day_of_week(self):
        days = [
            "Понедельник",
            "Вторник",
            "Среда",
            "Четверг",
            "Пятница",
            "Суббота",
            "Воскресенье"
        ]

        day_index = datetime(self.year, self.month, self.day).weekday()

        return days[day_index]

    def leap_year(self):
        if (self.year % 4 == 0 and self.year % 100 != 0) or (self.year % 400 == 0):
            return "Является Високосным годом ( True )"
        else:
            return "Не является Високосным годом ( False )"

    def days_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:
    regular = r"\( \w* \)"

    text = self.leap_year()

    match = re.search(regular, text)

    if match:
        a = match.group()

        del_replace = ["(", ")"]
        for i in del_replace:
            a = a.replace(i, "")

    else:
        print("Совпадение не найдено")

    if match.group():
        return 29
    else:
        return 28

def program():
    input_print = ["Напишите год: ", "Напишите месяц: ", "Напишите число: "]
    year, month, day = [int(input(input_print[i])) for i in range(3)]

    day1 = Calendar(year=year, month=month, day=day)

    print(
        f"\nДень недели: {day1.day_of_week()}\n"
        f"Вид года: {day1.leap_year()}\n"
        f"Количество дней в месяце: {day1.days_month()}\n"
    )

# program()

```

## Протокол работы программы:

Напишите год: 2024

Напишите месяц: 5

Напишите число: 20

День недели: Понедельник

Вид года: Является Високосным годом ( True )

Количество дней в месяце: 31

Process finished with exit code 0

## Задача Блока 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".

## Код программы:

```
"""
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, num_lap, color) -> None:
        self.species = species
        self.num_lap = num_lap
        self.color = color

class Dog(Animal):
    def __init__(self, species, num_lap, color, nickname, vid):
        super().__init__(species, num_lap, color)
        self.nickname = nickname
        self.vid = vid

    def __str__(self):
        return (f"\nВид: {self.species}\n"
                f"Количество Лап: {self.num_lap}\n"
                f"Цвет шерсти: {self.color}\n"
                f"Кличка: {self.nickname}\n"
                f"Порода: {self.vid}\n"
                )

pet = Dog(
    "Собака",
    4,
    "Белый с чёрными ушками",
    "Кутя",
    "Алабай"
)

print(pet)
```

## Протокол работы программы:

Вид: Собака

Количество Лап: 4

Цвет шерсти: Белый с чёрными ушками

Кличка: Кутя

Порода: Алабай

Process finished with exit code 0

### Задача Блока 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.

### Код программы:

```
"""
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

from random import randint
from task1 import Calendar

def save_def(data, name_file):
    with open(name_file, 'wb') as file:
        pickle.dump(data, file)

def load_def(name_file):
    with open(name_file, 'rb') as file:
        data = pickle.load(file)

    return data

day1, day2, day3 = (
    Calendar(2024, 5, 20),
    Calendar(2024, 6, 20),
    Calendar(2024, 7, 20)
)

save_def([day1, day2, day3], name_file='file.pkl')

load_file = load_def('file.bat')

for i in load_file:
    print(i)
```

### Протокол работы программы:

Год: 2024

Месяц: 05

Число: 20

Год: 2024

Месяц: 06

Число: 20

Год: 2024

Месяц: 07

Число: 20

Process finished with exit code 0

## **Вывод:**

В процессе выполнения практического занятия выработал навыки работы с ООП в IDE PyCharm.

Выполнены разработка кода, отладка, тестирование и оптимизация программного кода.

Готовые программные коды выложены на **GitHub**.  
( <https://github.com/Zagidin/Magamedragimov-IS-22> )