

Практическое занятие № 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 Community.

Задание 1:

Постановка задачи: 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.

Текст программы:

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

    def get_weekday(self):
        days = ['Понедельник', 'Вторник', 'Среда',
                'Четверг', 'Пятница', 'Суббота', 'Воскресенье']
        weekday = (self.day + 2 * self.month + 3 *
                    (self.month + 1) // 5 + self.year + self.year // 4 -
                    self.year // 100 + self.year // 400) % 7
        return days[weekday]

    def is_leap_year(self):
        if self.year % 4 == 0 and (self.year % 100 != 0
or self.year % 400 == 0):
            return True
        else:
            return False

    def get_days_in_month(self):
        days_in_months = [31, 28, 31, 30, 31, 30, 31,
31, 30, 31, 30, 31]

        days = days_in_months[self.month - 1]

        if self.is_leap_year() and self.month == 2:
```

```
        days += 1

    return days

calendar = Calendar(2024, 5, 7)
print(calendar.get_weekday())
print(calendar.is_leap_year())
print(calendar.get_days_in_month())
```

Протокол работы программы:

"C:/Users/Yana/Documents/vedenova is-22/PZ 16/pz16_1.py"
Вторник
True
31
Process finished with exit code 0

Задание 2:

Постановка задачи: Создайте класс "Животное", который содержит информацию о виде и возрасте животного. Создайте классы "Собака" и "Кошка", которые наследуются от класса "Животное" и содержат информацию о породе

Текст программы:

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

    def get_info(self):
        return f"Это {self.species} возрастом {self.age} лет."

class Pet(Animal):
    def __init__(self, species, age, breed):
        super().__init__(species, age)
        self.breed = breed

    def get_info(self):
        return f"Это {self.species} породы {self.breed} возрастом {self.age} лет."

dog = Pet("Собака", 5, "Лабрадор")
print(dog.get_info())
```

```
cat = Pet("Кошка", 6, "Сиамская")
print(cat.get_info())
```

Протокол работы программы:

Это Собака породы Лабрадор возрастом 5 лет.
Это Кошка породы Сиамская возрастом 6 лет.

Process finished with exit code 0

Задание 3:

Постановка задачи 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.

Текст программы:

```
import pickle

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

    def get_weekday(self):
        days = ['Понедельник', 'Вторник', 'Среда',
                'Четверг', 'Пятница', 'Суббота', 'Воскресенье']
        weekday = (self.day + 2 * self.month + 3 *
                    (self.month + 1) // 5 + self.year + self.year // 4 -
                    self.year // 100 + self.year // 400) % 7
        return days[weekday]

    def is_leap_year(self):
        if self.year % 4 == 0 and (self.year % 100 != 0
        or self.year % 400 == 0):
            return True
        else:
            return False

    def get_days_in_month(self):
        days_in_months = [31, 28, 31, 30, 31, 30, 31,
        31, 30, 31, 30, 31]

        days = days_in_months[self.month - 1]
```

```

        if self.is_leap_year() and self.month == 2:
            days += 1

    return days

def save_def(calendars, filename):
    with open(filename, 'wb') as f:
        pickle.dump(calendars, f)

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

calendar1 = Calendar(2024, 5, 7)
calendar2 = Calendar(2023, 12, 31)
calendar3 = Calendar(2022, 2, 14)

```

Протокол работы программы:

Вторник
True
31

Воскресенье
False
31

Пятница
False
28

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработала навыки составления программ с ООП в IDE PyCharm Community.
Готовые программные коды выложены на GitHub.