

Практическое занятие №16

1.Тема: составление программ с использованием ООП

2.Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с ООП в IDE PyCharm Community

3.Постановка задачи(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.

4. Тип алгоритма - циклический

Текст программы

```
import calendar
from datetime import datetime

class Calendar:
    def __init__(self):
        self.date_time = datetime.now()
    def weekday(self):
        days_of_week = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница',
                        'Суббота', 'Воскресенье']
        return days_of_week[self.date_time.weekday()]

    def leap_year(self):
        return calendar.isleap(self.date_time.year)

    def days_in_month(self):
        return calendar.monthrange(self.date_time.year, self.date_time.month)[1]

my_calendar = Calendar()

print("День недели:", my_calendar.weekday())
print("Високосный год:", my_calendar.leap_year())
print("Дни в месяце:", my_calendar.days_in_month())
```

Протокол работы программы:

День недели: Вторник

Високосный год: True

Дни в месяце: 31

Process finished with exit code 0

5. Постановка задачи(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".

6. Текст программы :

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

    def describe(self):
        print(f"Это {self.species} с {self.num_legs} лапами и с {self.fur_color} окрасом.")

class Dog(Animal):
    def __init__(self, species, num_legs, fur_color, name, breed):
        super().__init__(species, num_legs, fur_color)
        self.name = name
        self.breed = breed

    def describe(self):
        super().describe()
        print(f"{self.name} - это {self.breed}.")

dog = Dog("Собака", 4, "коричневый", "Дружок", "Лабрадор")
dog.describe()
```

7. Протокол работы программы:

Это Собака с 4 лапами и с коричневый окрасом.

Дружок - это Лабрадор.

Process finished with exit code 0

8. 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

9. Текст программы:

```
import calendar
from datetime import datetime
import pickle

class Calendar:
    def __init__(self):
        self.date_time = datetime.now()
    def weekday(self):
        days_of_week = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница',
                        'Суббота', 'Воскресенье']
        return days_of_week[self.date_time.weekday()]

    def leap_year(self):
        return calendar.isleap(self.date_time.year)

    def days_in_month(self):
        return calendar.monthrange(self.date_time.year, self.date_time.month)[1]

def save_def(calendar_objects, filename):
    """Сохраняет экземпляры класса Calendar в файл"""
    with open(filename, "wb") as f:
        pickle.dump(calendar_objects, f)

def load_def(filename):
    """Загружает экземпляры класса Calendar из файла"""
    with open(filename, "rb") as f:
        calendar_objects = pickle.load(f)
    return calendar_objects

calendar1 = Calendar()
calendar2 = Calendar()
calendar3 = Calendar()
save_def([calendar1, calendar2, calendar3], "calendar.data")
```

```
for i in load_def("calendar.data"):
    print(i.weekday())
```

10. Код выполнения программы :

Вторник

Вторник

Вторник

Process finished with exit code 0

11. Вывод : Сегодня я составлял программы с использованием ООП, также закреплял усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с ООП в IDE PyCharm Community.