

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 ООП выражений в IDE PyCharm Community

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

Текст 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 datetime
class Callendar:
    def __init__(self, y, m, d):
        self.y = y
        self.m = m
        self.d = d

    def find_day_of_week(self):
        week_days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday"]
        date_obj = datetime.date(self.y, self.m, self.d)
        week_day_num = date_obj.weekday()
        return week_days[week_day_num]

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

    def days_in_month(self):
        if self.m in [1, 3, 5, 7, 8, 10, 12]:
            return 31
        elif self.m in [4, 6, 9, 11]:
            return 30
        else:
            if self.is_leap_year():
                return 29
            else:
                return 28
```

Протокол работы:

```
C:\ger\.venv\Scripts\python.exe C:\ger\PZ_16\PZ_16_1.py
|
Process finished with exit code 0
```

№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_paws, fur_color):
        self.species = species
        self.num_legs = num_paws
        self.fur_color = fur_color

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

# Пример использования классов
dog1 = Dog("Собака", 4, "рыжий", "Барон", "Лабрадор")
```

Протокол работы:

```
C:\ger\.venv\Scripts\python.exe C:\ger\PZ_16\PZ_16_2.py
|
Process finished with exit code 0
```

№ 3

для задачи из блока 1 создать две функции, save_def и load_def, которые позволяют сохранить 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:

```
# 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_paws, fur_color):
        self.species = species
```

```
        self.num_legs = num_paws
        self.fur_color = fur_color

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

# Пример использования классов
dog1 = Dog("Собака", 4, "рыжий", "Барон", "Лабрадор")
```

Протокол работы

```
C:\ger\.venv\Scripts\python.exe C:\ger\PZ_16\PZ_16_3.py
|
Process finished with exit code 0
```

В процессе выполнения задания выработал основные принципы составления программ, приобрел навыки составления программ с использованием регулярных выражений в IDE PyCharm Community

Готовые решения выложены на Github