

Практическое занятие № 16

Тема: Составление программ с в IDE PyCharm Community.

Цели: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с ООП в IDE PyCharm Community.

Постановка задачи №1:

Создайте класс "Здание" с атрибутами "адрес" и "количество этажей".
Напишите метод, который выводит информацию о здании в формате "Адрес: адрес, Количество этажей: этажи".

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

Тип алгоритма: линейный

Текст программы:

```
# Создайте класс "Здание" с атрибутами "адрес" и "количество этажей".
# Напишите метод, который выводит информацию о здании в формате
# "Адрес: адрес, Количество этажей: этажи".

# 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

1 usage
def save_def(filename, *objects, _):
    # Сохраняем объекты в файл в бинарном формате

    with open(filename, "wb") as f:
        pickle.dump(objects, f)

1 usage
def load_def(filename):
    # Загружаем объекты из файла в бинарном формате

    with open(filename, "rb") as f:
        return pickle.load(f)

3 usages
class Building:
    def __init__(self, address, number_of_floors):
def()
```

```
3 usages
class Building:
    def __init__(self, address, number_of_floors):
        self.address = address
        self.number_of_floors = number_of_floors

    @staticmethod
    def conclusion(text, count):
        return f"{text} {count}"

    def __str__(self):
        return f"Адрес: {self.address}, Количество этажей: {self.number_of_floors}"

# Создание экземпляров класса
building1 = Building(input("Введите адрес №1: "), input("Введите количество этажей: "))
building2 = Building(input("Введите адрес №2: "), input("Введите количество этажей: "))
building3 = Building(input("Введите адрес №3: "), input("Введите количество этажей: "))
save_def( filename: "buildings.pkl", *objects: building1, building2, building3)
buildings = load_def("buildings.pkl")

for building in buildings:
    print("\n", building)
```

Протокол работы программы:

Введите адрес №1: Россия, республика Татарстан, г.Казань, улица Братьев Касимовых, 64.

Введите количество этажей: 3

Введите адрес №2: Россия, республика Татарстан, г.Казань, улица Братьев Касимовых, 65.

Введите количество этажей: 5

Введите адрес №3: Россия, республика Татарстан, г.Казань, улица Братьев Касимовых, 63.

Введите количество этажей: 5

Адрес: Россия, республика Татарстан, г. Казань, улица Братьев Касимовых, 64.,
Количество этажей: 3

Адрес: Россия, республика Татарстан, г.Казань, улица Братьев Касимовых, 65.,
Количество этажей: 5

Адрес: Россия, республика Татарстан, г.Казань, улица Братьев Касимовых, 63.,
Количество этажей: 5

Process finished with exit code 0

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".

Тип алгоритма: линейный

Текст 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".  
2 usages  
class Animal:  
    def __init__(self, view, number_of_paws, coat_color):  
        self.view = view  
        self.number_of_paws = number_of_paws  
        self.coat_color = coat_color  
  
1 usage  
class Dog(Animal):  
    def __init__(self, view, number_of_paws, coat_color, moniker, breed):  
        super().__init__(view, number_of_paws, coat_color)  
        self.moniker = moniker  
        self.breed = breed  
  
animal = Animal( view: "Кошка", number_of_paws: 4, coat_color: "Черный")  
dog = Dog( view: "Собака", number_of_paws: 4, coat_color: "Коричневый", moniker: "Рекс", breed: "Пудель")  
  
print(dog.view)  
print(dog.moniker)
```

Протокол работы программы:

Собака

Рекс

Process finished with exit code 0

Вывод: В процессе выполнения практической работы я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ с ООП в IDE PyCharmCommunity.

Студентка группы ИС-23 Камилова А.Н.

Студентка группы ИС-23 Камилова А.Н.