

Практическое занятие №16

Тема: составление программ с использованием ООП 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 «Счетчик», который имеет атрибут текущего значения и методы для инкремента и декремента значения.
2. 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
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

Текст программы:

```
1  #Создайте класс «Счетчик», который имеет атрибут текущего значения и методы для
2  #инкремента и декремента значения
3  class Counter:
4      def __init__(self, initial_value=0):
5          self.value = initial_value
6      def increment(self):
7          self.value += 1
8      def decrement(self):
9          self.value -= 1
10 counter = Counter(5)
11 print("Текущее значение:", counter.value)
12 counter.increment()
13 print("После инкремента:", counter.value)
14 counter.decrement()
15 print("После декремента:", counter.value)
```

2.

```
1  #Создайте класс "Животное", который содержит информацию о виде и возрасте
2  #животного. Создайте классы "Собака" и "Кошка", которые наследуются от кл
3  #"Животное" и содержат информацию о породе.
   2 usages  victoriaskobelina
4  class Animal:
   victoriaskobelina
5      def __init__(self, species, age):
6          self.species = species
7          self.age = age
8
   1 usage  victoriaskobelina
9  class Dog(Animal):
   victoriaskobelina
10     def __init__(self, species, age, breed):
11         super().__init__(species, age)
12         self.breed = breed
13
   1 usage  victoriaskobelina
14  class Cat(Animal):
   victoriaskobelina
15     def __init__(self, species, age, breed):
16         super().__init__(species, age)
17         self.breed = breed
18
19  dog = Dog( species: "Собака", age: 3, breed: "Доберман")
20  cat = Cat( species: "Кошка", age: 5, breed: "Манчкин")
21
22  print("Вид собаки:", dog.species)
23  print("Возраст собаки:", dog.age)
24  print("Порода собаки:", dog.breed)
25
26  print("Вид кошки:", cat.species)
27  print("Возраст кошки:", cat.age)
28  print("Порода кошки:", cat.breed)
```

3.

```

1  #Для задачи из блока 1 создать две функции, save_def и load_def, которые позвол
2  #сохранять информацию из экземпляров класса (3 шт.) в файл и загружать ее обрат
3  #Использовать модуль pickle для сериализации и десериализации объектов Python в
4  #бинарном формате
5  import pickle
6  3 usages new *
7  class Counter:
8      new *
9      def __init__(self, value=0):
10         self.value = value
11     3 usages new *
12     def increment(self):
13         self.value += 1
14     new *
15     def decrement(self):
16         self.value -= 1
17  1 usage new *
18  def save_def(filename, *counters):
19     with open(filename, 'wb') as file:
20         pickle.dump(counters, file)
21  1 usage new *
22  def load_def(filename):
23     with open(filename, 'rb') as file:
24         counters = pickle.load(file)
25     return counters
26
27  counter1 = Counter(5)
28  counter2 = Counter(10)
29  counter3 = Counter(15)
30
31  counter1.increment()
32  counter2.increment()
33  counter3.increment()
34
35  save_def( filename: 'counters.pickle', *counters: counter1, counter2, counter3)
36  loaded_counters = load_def('counters.pickle')
37
38  for c in loaded_counters:
39      print("Значение после инкремента: ", c.value)

```

Протокол работы программы:

1. Текущее значение: 5
После инкремента: 6
После декремента: 5
Process finished with exit code 0
Программа успешно завершена!
2. Вид животного: Собака

Возраст собаки: 3

Порода собаки: Доберман

Вид животного: Кошка

Возраст кошки: 5

Порода кошки: Манчкин

Process finished with exit code 0

Программа успешно завершена!

3. Значение после инкремента: 6

Значение после инкремента: 11

Значение после инкремента: 16

Process finished with exit code 0

Программа успешно завершена!

Вывод: в процессе выполнения практического занятия были выработаны навыки составления программ с использованием ООП в IDE PyCharm Community и закреплены усвоенные знания, понятия, алгоритмы, основные принципы составления программ. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.