

# Контрольная по Python



# Задача №1

Создайте метакласс ClassStorer. Будучи приписан к классу этот метакласс позволяет загружать сохраненные объекты из файла и сохранять объекты в файл. Если не удалось найти файл, то необходимо сформировать исключение LoadError.

При попытке провести загрузку из файла соответствующего другому классу тоже должно быть сгенерировано исключение (InvalidClassStorageError).

Загрузка должна производиться через вызов конструктора с параметром строкового типа.

Сохранение через метод store с параметром – строкой. В сохраненном файле должны быть сохранены все поля класса.

Например:

```
class A(metaclass = ClassStorer):  
    pass
```

```
>>>a = A()  
>>>a.x = 3  
>>>a.store("test.txt")  
>>>b = A("test.txt")
```

# Задача №2

В данном задании запрещается пользоваться библиотечными функциями!

- Напишите класс двоичной кучи. Куча должно поддерживать операции сложения (+) с другой кучей или с числом. При попытке сложить с другими типами должно выдаваться исключение. Результатом сложения является НОВАЯ куча, а не измененная старая.
- Операция унарного минуса должна извлекать корень из кучи и возвращать его значение.
- Помимо этого необходимо реализовать возможность итерироваться по куче в порядке увеличения ее элементов. При этом куча не должна изменяться!

```
for i in range(100):  
    heap = heap + i  
top = -heap  
for i in iter(heap):  
    print(i)
```

## Задача №3

Напишите декоратор `@spy` и функцию `bond()`, которая является генератором и выдает кортежи из двух элементов – **суммарного** времени работы **продекоррированной** с помощью `@spy` функции и ее имени. Результаты должны быть отсортированы в порядке убывания времени работы.

# Задача №4

Ваша задача – вывести дерево вызовов функций. Не забудьте про рекурсию...

```
foo2(1)
foo(2)
printStack()
```

Пример:

```
def test1(num):
    return (num == 1)
def test2(num):
    return (num > 3)
def foo(num):
    print("Original")
    test1(num)
def foo2(num):
    foo(num)
    test1(num)
```

ВЫВОДИТ:

```
....
foo2(1)
    foo(1)
        test1(1)
            test1(1)
foo(2)
    test1(2)
```