

Спазвайте практиките за обектно-ориентирано програмиране, коментирани на упражнения и лекции.

Задача 1 (3pts)

Може да се използва `<cstring>`

Напишете собствена имплементация на класа `String`, която да бъде с име `MyString`. `MyString` трябва да може да съдържа текст и да поддържа инициализиране с низ, добавяне на символ в края на низа, принтиране, както и оператори за сравнение (`==`, `!=`, `>`, `<`) (подобна на `strcmp`)

Примерен `main()`:

```
MyString temp("Hello world");
temp.print();

temp.append('!');
temp.print();
```

Примерен изход:

```
Hello world
Hello world!
```

Задача 2 (3pts)

Напишете клас `IntCounter`, който ще съхранява динамично зададена `int` променлива.

`IntCounter` има следните свойства:

- Съдържа броя на "референциите" към тази променлива (всяка различна променлива, има различен брояч)
- При инициализация, приема указател - стойността, която ще се пази
- При създаване на копие на `IntCounter`, се увеличава броя на референциите
- Метод, който връща стойността на пазената променлива
 - При унищожаване на `IntCounter`, брояча се намалява. Ако брояча стигне до 0, то стойността която се пази, се унищожава

Примерен `main()`:

```

int* some_number = new int(5);
IntCounter first(some_number);
std::cout << first.get_count() << std::endl;
IntCounter second = first;
std::cout << first.get_count() << std::endl;

{
    IntCounter third(second);
    std::cout << first.get_count() << std::endl;
}
std::cout << first.get_count() << std::endl;

```

Примерен изход:

```

1
2
3
2

```

Примерен `main()`:

```

int* some_number = new int(5);
IntCounter first(some_number);

int* some_other_number = new int(7);
IntCounter second(some_other_number);

std::cout << first.get_count() << std::endl;
std::cout << second.get_count() << std::endl;

```

Примерен изход:

```

1
1

```

Задача 3 (4pts)

Моделирайте опростена версия на `SVG` (Scalable Vector Graphics), която да поддържа визуализация на правоъгълници, дефинирани чрез две точки (лежащи на двата края на диагонала на правоъгълника), в двумерното пространство. `SVG` да има следните функционалности:

- Добавяне на нов правоъгълник
- Създаване на правоъгълник по две точки и добавяне към списък с правоъгълници
- Извеждане индекса на правоъгълника, който има най-голямо лице, сред всички правоъгълници от гореспоменатия списък.