

3.7 Конструктор копирования и оператор присваивания

Шаг 9

Компилятору C++ позволено выполнять оптимизации, связанные с удалением ненужных вызовов конструктора копирования (copy elision) при определенных условиях, даже если конструктор копирования содержит сайд эффекты (например, вывод в `std::cout`), поэтому на большинстве современных компиляторов конструктор копирования не будет вызван ни в одной из указанных ситуаций. Однако, компилятор не обязан выполнять эту оптимизацию.

В каких из приведенных ниже ситуаций **может быть вызван** конструктор копирования класса `String`:

1.

```
String spaces(size_t n) {
    const String s(n, ' ');
    return s;
}

int main() {
    std::cout << spaces(10).str << "\n";
    return 0;
}
```

2.

```
int main() {
    String ten_spaces;
    ten_spaces = String(10, ' ');
    std::cout << ten_spaces.str << "\n";
    return 0;
}
```

3.

```
void foo(String str) {  
    std::cout << str.str << "\n";  
}  
  
int main() {  
    foo(String(10, ' '));  
    return 0;  
}
```

4.

```
void bar(const String &str) {  
    std::cout << str.str << "\n";  
}  
  
int main() {  
    bar("          "); // ten spaces;  
    return 0;  
}
```

Чтобы решить это задание откройте
<https://stepik.org/lesson/550/step/9>