

# BreakTheCode

---

В этой задаче вам предстоит сломать инкапсуляцию.

---

А именно, в систему загружен следующий код:

```
class SecretClass {  
private:  
    std::string token;  
protected:  
    void SetTokenTo(SecretClass& another) {  
        another.token = token;  
    }  
public:  
    SecretClass(const std::string& token) : token(token) {};  
    std::string GetToken() const {  
        return token;  
    }  
};  
  
void externalFunction(SecretClass& secret);  
  
int main() {  
    SecretClass secret("FUTURE");  
    externalFunction(secret);  
    assert(secret.GetToken() == "CODE");  
}
```

`assert` работает следующим образом. Если выражение в скобках является `true`, то ничего не происходит. Если выражение в скобках является `false`, ваше решение падает с ошибкой `RE`.

---

Ваша задача реализовать функцию

```
void externalFunction(SecretClass& secret);
```

таким образом, чтобы выражение `secret.GetToken() == "CODE"` в конце `main` в скобках `assert` было `true`.

Вы можете помимо этой функции реализовать и другие вспомогательные функции/классы, если они помогут вам решить задание. Весь ваш код будет вставлен в систему между классом `SecretClass` и функцией `main`.

---

В систему отправляйте лишь код функции, необходимые библиотеки и вспомогательные функции/классы. Всё остальное будет подключено автоматически

