Конструктор перемещения

30 мая 2017 г.

Выражения: lvalue vs. rvalue

lvalue

- объект **с именем**
- обычно существует за пределами одного выражения
- может быть **и справа, и слева** в выражении
- можно получить **адрес**

rvalue

- **безымянный** объект
- обычно – **временное значение** (не сохраняется за пределами того выражения, где используется)
- всегда находится **справа**
- **нельзя** получить адрес

Выражения: lvalue vs. rvalue

```
int x = 3 + 4;
const int y = 1;
int z = x + y;

int i = 1;
(double) i;
7 = i;                  // Error
i * 4 = 7;              // Error

int f( ) { return 10; }
i = f( );
f( ) = i;               // Error
```

Семантика перемещения (move-семантика)

1. Позволяет создавать код, который переносит ресурсы из одного объекта в другой.
2. Позволяет переносить ресурсы из временных объектов, на которые невозможно ссылаться из других мест в программе.
3. Обычно повышается производительность кода.
4. Помогает там, где нельзя копировать объект, но нужно передать права на него.

Конструктор перемещения

1. Может генерироваться компилятором автоматически.
2. Если явно определен деструктор или конструктор копирования, то по умолчанию конструктор перемещения сгенерирован не будет.

Конструктор перемещения

```
X(X&& myObj);
```

где X – имя класса

- принимает неконстантную ссылку
- в объявлении обязательно нужны &&
- не возвращает значение
- данные забираются у передаваемого объекта
- в поля передаваемого объекта устанавливаются значения по умолчанию

Конструктор перемещения

X(const X& obj);	X(X&& obj);

string(const string& s) :
  buf(new char[s.size + 1]) ,
  size(s.size)
{
  strcpy(buf, s.buf);
}
	string(string&& s) :
	  buf(s.buf),
	  size(s.size)
	{
	  s.buf = nullptr;
	  s.size = 0;
	}

Вызов конструктора перемещения

- компилятор сам решает, какой из конструкторов (копирования или перемещения) вызвать
- чтобы явно указать компилятору, что нужно вызвать именно конструктор перемещения, имя вызываемого объекта передается в функцию std::move

```
MyClass m1;
MyClass m2 = std::move(m1);
```

Задание

Выберите верные объявления конструкторов.

1) MyClass();
2) MyClass(const int x = 1; const int y = 10);
3) MyClass(const int x = 1);
4) MyClass(const int);
5) MyClass(const std::string&);
6) MyClass(const MyClass&);
7) MyClass(const MyClass&&);
8) MyClass(MyClass&);
9) MyClass(MyClass&&);

Литература

• stackoverflow.com/questions/3106110/what-are-move-semantics

•Мейерс С. Эффективный и современный C++. Глава 5.

Вопросы?