C++. Базовый уровень

Лямбда-выражения в C++. Создание и использование лямбда-выражений. Захват переменных в лямбда-выражениях.

КОД будущего

Минцифры России

инжиниум МГТУ им. Н.Э. Баумана

Национальные проекты России

ЦИФРОВАЯ ЭКОНОМИКА

20.35 УНИВЕРСИТЕТ

2023

Лямбда-выражения

Ключевое слово auto в C++ используется для автоматического вывода типа переменной на основе ее инициализатора. Это позволяет упростить код и избежать ошибок, связанных с неправильным указанием типа

Для объявления локальных переменных, которые инициализируются непосредственно или с помощью возвращаемого значения функции.

```
auto x = 42; // x имеет тип int
auto y = 3.14; // y имеет тип double
auto z = x + y; // z имеет тип double
auto s = "hello"; // s имеет тип const char*
```

Лямбда-выражения

```
// Функция, которая возвращает максимальный из двух аргументов
    template <typename T>
    T max(T a, T b) {
        return (a > b) ? a : b;
    }
    // Функция, которая возвращает максимальный из двух аргументов с
использованием auto
    auto max(auto a, auto b) {
        return (a > b) ? a : b;
    }
    // Вызов функции max с разными типами аргументов
    cout << max(10, 20) << endl; // int
    cout << max(2.5, 3.7) << endl; // double
    cout << max("apple", "banana") << endl; // string
```

Лямбда-выражения

Лямбда-выражение состоит из списка параметров, тела функции и необязательного списка захвата.

```
[capture_list](parameters) -> return_type {
    // тело функции
}
```

[] (параметры) { действия }

Лямбда-выражения

- capture list - это список переменных, которые будут доступны внутри лямбда-выражения. Он может быть пустым, содержать одну или несколько переменных, разделенных запятыми, или использовать символ = для захвата всех переменных по значению или символ & для захвата всех переменных по ссылке.

- parameter list - это список параметров, которые принимает лямбда-выражение. Он может быть пустым, содержать один или несколько параметров, разделенных запятыми, или использовать символ ... для принятия произвольного количества аргументов.

- return type - это тип возвращаемого значения лямбда-выражения. Он может быть опущен, если лямбда-выражение возвращает void или если тип возвращаемого значения может быть выведен из тела функции.

Лямбда-выражения

[]() { std::cout << "Hello" << std::endl; }

```
#include <iostream>

int main()
{
    []() {std::cout << "Hello" << std::endl; } ();
    // или так
    [] {std::cout << "Hello" << std::endl; } ();
}
```

Лямбда-выражения

```
#include <iostream>

int main()
{
    auto print{ [](const std::string& text) {std::cout << text << std::endl; } };

    // вызываем лямбда-выражение
    print("Hello World!");          // Hello World!
    print("Good bye, World...");    // Good bye, World...
}
```

Лямбда-выражения

```
#include <iostream>

int main()
{
    auto sum{ [](int a, int b) {return a + b; } };

    // вызываем лямбда-выражение
    std::cout << sum(10, 23) << std::endl;  // 33

    // присваиваем его результат переменной
    int result{ sum(1, 4) };
    std::cout << result << std::endl;       // 5
}
```

Лямбда-выражения

```
#include <iostream>

void do_operation(int a, int b, int (*op)(int, int))
{
    std::cout << op(a, b) << std::endl;
}

int main()
{
    auto sum{ [](int a, int b) {return a + b; } };
    auto subtract{ [](int a, int b) {return a - b; } };

    do_operation(10, 4, sum);           // 14
    do_operation(10, 4, subtract);     // 6
}
```

Лямбда-выражения

```
#include <iostream>

int main()
{
    auto add = [](auto a, auto b) {return a + b; };
    //auto print = [](const auto& value) {std::cout << value << std::endl; };

    std::cout << add(2, 3) << std::endl;        // 5 - складываем числа int
    std::cout << add(2.2, 3.4) << std::endl;    // 5.6 - складываем числа double

    std::string hello{ "hello " };
    std::string world{ "world" };
    std::cout << add(hello, world) << std::endl;    // hello world - складываем строки
}
```

Лямбда-выражения

1. Напишите лямбда-выражение, которое принимает целое число и возвращает его квадрат.

2. Напишите лямбда-выражение, которое принимает два булевых значения и возвращает их логическое ИЛИ. Например, для значений true и false лямбда-выражение должно вернуть true.