МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность Программная инженерия

**РЕФЕРАТ**

по теме:

«Отличия между перегрузкой и переопределением функции»

Выполнил:

Студент 1 курса 10 группы, 2 подгруппы

Специальность ПИ Мандрик Алексей Иванович

Проверил:

Белодед Николай Иванович

Минск 2024

**Оглавление**

Предисловие……………………………………………………………………….3

Перегрузка функций………………………………………………………………3

Переопределение функций……………………………………………………….5

Вывод………………………………………………………………………………6

**Предисловие**

На лекции преподаватель предложил рассмотреть отличия между перегрузкой и переопределением функции в С++. В данном реферате мы изучим, что такое перегрузка и переопределение функции, а также рассмотрим основные различия между ними.

**Перегрузка функции (Function Overloading)**

Перегрузка функций в языке С++ это возможность одновременно использовать в одной области одноимённых функций, но с разным количеством передаваемых параметров. Компилятор ещё на этапе компиляции выбирает на основе передаваемых параметров нужную функцию и в конкретном случае использует именно её.

**Пример кода**

#include <iostream>

using namespace std;

class name

{

public:

int Sum(int a, int b) //Функция с двумя параметрами int

{

return a + b;

}

int Sum(int a, int b, int c) //Функция с тремя параметрами int

{

return a + b + c;

}

};

int main()

{

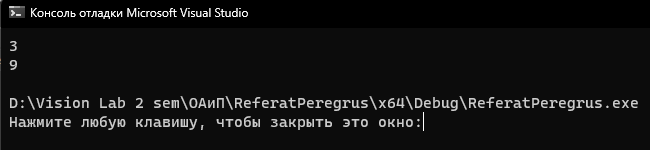
name n;

cout << n.Sum(1, 2) << endl; // вызов Sum(int, int)

cout << n.Sum(1, 3, 5) << endl; // вызыв Sum(int, int, int)

return 0;

}

**Результат выполнения программы**

**Дополнение**

Существует возможность одновременно использовать в одной области несколько одноимённых функций с одинаковым количеством передаваемых параметров. Для этого необходим разный тип данных функций.

**Код программы**

#include <iostream>

using namespace std;

class name

{

public:

int Sum(int a, int b) //Функция с двумя параметрами int

{

return a + b;

}

double Sum(double a, double b) //Функция с двумя параметрами double

{

return a + b;

}

};

int main()

{

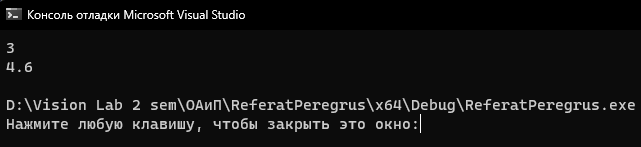
name n;

cout << n.Sum(1, 2) << endl; // вызов Sum(int, int)

cout << n.Sum(1.2, 3.4) << endl; // вызыв Sum(double, double)

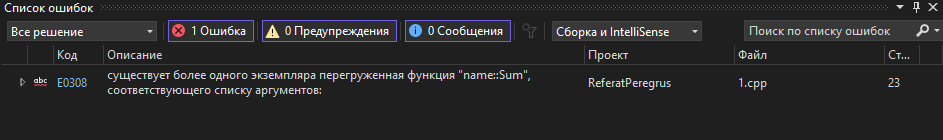
return 0;

}

**Результат выполнения программы****

**Замечание!!!**

**НЕДОПУСТИМО использовать РАЗНЫЕ типы данных, которые, которые занимают ОДИНАКОВЫЙ размер памяти. Если в программе использовать вместо типа double (8 байт) тип float (4 байта), то появится соответствующая ошибка, т.к. int (4 байта)**

******

**Переопределение функций (Function Overriding)**

Переопределение функций происходит, когда подкласс переопределяет метод родительского класса. Основной задачей переопределения является изменение поведения метода в подклассе. Производный класс может переопределить поведение виртуальной функции.

**Код программы**

#include <iostream>

using namespace std;

class name //Родительский класс

{

public:

virtual int Sum(

{

cout << "Before function" << endl;

return 0;

}

};

class Overriding: public name //Переопределённый класс

{

public:

int Sum() override

{

cout << "After function" << endl;

return 0;

}

};

int main()

{

name n;

name\* namePtr = &n;

namePtr->Sum(); // Вывод: "Before function"

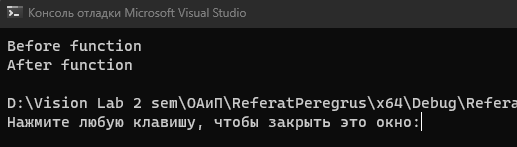
Overriding o;

namePtr = &o;

namePtr->Sum(); // Вывод: "After function"

return 0;

}

**Результат выполнения программы** ****

В данном примере класс Overriding наследуется от класса name, и оба класса определяют функцию Sum().

**Вывод**

Хоть два способа и выполняют одну функцию, они имеют различный принцип действия. Перегрузка добавляет разнообразие методов в одном классе, в то время как переопределение даёт возможность менять поведение методов в подклассах.

***Спасибо за внимание! Удачи!***