МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Национальный исследовательский ядерный университет «МИФИ»

**Институт интеллектуальных кибернетических систем**

**кафедра “Кибернетика”(22)**

“Объектно-Ориентированное Программирование”

**Лабораторная работа по ООП:**

«Лабораторная работа № 1. Изучение базовых конструкций языка C++»

*Выполнил: Локтионов А.А.. гр. М19-514*

*Преподаватель: Шапкин П.А.*

Москва

2020 г.

**Задание 1.** Написать программы, использующие функции потокового ввода-вывода:

• с полным указанием пространства имен std;

void IOWithFullNamespace() {

std::string input;

std::cin >> input;

std::cout << input << std::endl;

}

• с сокращенным указанием пространства имен (использовать using);

using namespace std;

void IOWithoutNamespace() {

string input;

cin >> input;

cout << input << endl;

}

• вызывающую ошибку компиляции из-за отсутствия указания пространства имен.

void IOCompileError() {

string input;

cin >> input;

cout << input << endl;

}

**Задание 2.** Написать программу, определяющую различные варианты функции plus: обычное сложение и сложение по модулю 2. Использовать механизм пространств имен.

namespace real {

int plus(int a, int b) {

return a + b;

}

}

namespace ring2 {

int plus(int a, int b) {

return (a + b) % 2;

}

}

**Задание 3.** Написать процедуру, изменяющую значение своего аргумента, используя:

• передачу параметра по значению;

void ChangeArgument(string a) {

a = "Argument changed!";

}

• передачу параметра по ссылке;

void ChangeReferenceArgument(string& a) {

a = "Argument changed be reference!";

}

• указатели;

void ChangePtrArgument(string\* a) {

\*a= "Argument changed be prt!";

}

**Задание 4.** Проведите компиляцию кода листинга 1 и проанализируйте выполнение программы инструментом Dr. Memory.

CppCheck отчет:

Возврат указателя на локальную переменную:

[Task4.cpp:20] ->[Task4.cpp:19] ->[Task4.cpp:22]: (error)Returning pointer to local variable 'i' that will be invalid when returning.

Создание указателей без последующей очистки памяти

[Task4.cpp:11] : (error)Memory leak : i

[Task4.cpp:15] : (error)Memory leak : i

[Task4.cpp:17] : (error)Memory leak : i

Исправленная версия:

#include <stdlib.h>

#include <iostream>

#include <memory>

using namespace std;

void fv() {

int i = int(123);

cout << "i = " << i << endl;

}

void fp1() {

auto i = std::make\_shared<int>(123);

cout << "\*i = " << \*i << endl;

}

void fp2() {

auto i = std::make\_shared<int>(123);

cout << "\*i = " << \*i << endl;

i = std::make\_shared<int>(456);

cout << "\*i = " << \*i << endl;

}

int\* fr() {

static int i = int(123);

int\* ip = &i;

cout << "i = " << i << endl;

return ip;

}

int main() {

fv();

fp1();

fp2();

int\* ifr = fr();

cout << "\*ifr = " << \*ifr << endl;

return 0;

}