ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»

(СПбГУТ)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Факультет Информационных систем и технологий

Кафедра Безопасности информационных систем

**Дисциплина «Технологии программирования»**

**ОТЧЕТ**

**за практическое занятие №2**

**Тема: Взаимодействие функций в программах**

Выполнил

Студент 1 курса, гр. ИСТ-262

\_\_\_\_\_\_\_\_\_\_\_\_\_ Шибаев А. П.

Принял

Ассистент кафедры БИС

**\_\_\_\_\_\_\_\_\_\_\_\_\_** Новиков Е.А.

**Цель работы:** получение практических навыков проектирования взаимодействия функций в программах с использованием локальных и глобальных переменных.

**Постановка задачи: Задание 1.** Ввести с клавиатуры (согласно номеру в списке учебной группы, таблица 2) данные указанного типа (см. таблица 1), через указатели передать их в отдельную функцию, вычислить их сумму и определить размер отведённой для неё памяти. Полученную сумму привести явным способом к новому типу данных и вывести на экран её размер.

**Задание 2.** Заполнить пятиэлементный массив с клавиатуры и определить Вычисляемый показатель (см. таблица 1).

Каждое задание оформить в виде отдельной функции. В главной функции задавать исходные данные и выводить результаты на экран. Обмен данных между функциями выполнить в двух вариантах: 1) с использованием локальных переменных и 2) с использованием глобальных переменных

Программу спроектировать в виде меню, в котором по желанию пользователя обеспечить:

1. неоднократное выполнение любого задания;
2. выход из программы.

**Код программы**

#include <iostream>  
#include <limits>  
  
template<class T> void getUserInput(T &, char const \*);  
  
void firstTask(short \*, float \*);  
void secondTask(bool \*, const int \*);  
  
float calcSum(float \*, short \*);  
void getArrayFromUser(bool \*, int);  
int countFalse(bool \*, int);  
  
short data\_short\_global;  
float data\_float\_global;  
  
const int array\_size = 5;  
bool array\_global[array\_size];  
  
int main()  
{  
 float data\_float;  
 short data\_short;  
 float \*data\_float\_ptr;  
 short \*data\_short\_ptr;  
  
 const int \* array\_size\_ptr = &array\_size;  
 bool array[array\_size];  
  
 bool user\_quit = false;  
 char user\_menu\_choice;  
  
 while (!user\_quit)  
 {  
 std::cout << "1. Задание 1" << "\n";  
 std::cout << "2. Задание 2" << "\n";  
 std::cout << "3. Выход" << "\n";  
 getUserInput(user\_menu\_choice, "Ввод: ");  
  
 switch (user\_menu\_choice)  
 {  
 case '1':  
 std::cout << "Локальные переменные: " << "\n";  
 getUserInput(data\_short, "Short: ");  
 getUserInput(data\_float, "Float: ");  
 data\_short\_ptr = &data\_short;  
 data\_float\_ptr = &data\_float;  
 firstTask(data\_short\_ptr, data\_float\_ptr);  
  
 std::cout << "Глобальные переменные" << "\n";  
 getUserInput(data\_short\_global, "Global short: ");  
 getUserInput(data\_float\_global, "Global float: ");  
 data\_short\_ptr = &data\_short\_global;  
 data\_float\_ptr = &data\_float\_global;  
 firstTask(data\_short\_ptr, data\_float\_ptr);  
  
 break;  
 case '2':  
 std::cout << "Локальные переменные: " << "\n";  
 getArrayFromUser(array, array\_size);  
 secondTask(array, array\_size\_ptr);  
  
 std::cout << "Глобальные переменные" << "\n";  
 getArrayFromUser(array\_global, array\_size);  
 secondTask(array\_global, array\_size\_ptr);  
 break;  
 case '3':  
 user\_quit = true;  
 break;  
 default:  
 std::cout << "Некорректный ввод. Попробуйте еще раз" << "\n";  
 break;  
 }  
 system("clear");  
 }  
  
 return 0;  
}  
  
template<class T> void getUserInput(T &data, char const \* message)  
{  
 std::cout << message;  
 while(!(std::cin >> data))  
 {  
 std::cout << "Некорректный ввод. Попробуйте еще раз" << std::endl;  
 std::cin.clear();  
 std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');  
 std::cout << message;  
 }  
}  
  
void firstTask(short \* data\_short\_ptr, float \*data\_float\_ptr)  
{  
 std::cout << "Сумма: " << calcSum(data\_float\_ptr, data\_short\_ptr) << "\n";  
 std::cout << "Размер памяти под сумму: "  
 << sizeof(calcSum(data\_float\_ptr, data\_short\_ptr)) << "\n";  
 std::cout << "Рарзмер памяти под сумму после приведения к типу: "  
 << sizeof((long int)calcSum(data\_float\_ptr, data\_short\_ptr)) << "\n";  
}  
  
float calcSum(float \* data\_float, short \* data\_short)  
{  
 return \*data\_float + (float)\*data\_short;  
}  
  
void secondTask(bool \*array, const int \*arr\_size)  
{  
 std::cout << "Количество false: " << countFalse(array, \*arr\_size) << "\n";  
}  
  
void getArrayFromUser(bool \*array, int arr\_size)  
{  
 int user\_input;  
 bool user\_input\_is\_correct;  
  
 std::cout << "1 - true, 0 - false" << "\n";  
 for(int i = 0; i < arr\_size; i++)  
 {  
 std::cout << "a[" << i << "] = ";  
 do  
 {  
 getUserInput(user\_input, "");  
 switch (user\_input)  
 {  
 case 1:  
 array[i] = true;  
 user\_input\_is\_correct = true;  
 break;  
 case 0:  
 array[i] = false;  
 user\_input\_is\_correct = true;  
 break;  
 default:  
 std::cout << "Некорректный ввод. Попробуйте еще раз" << "\n";  
 user\_input\_is\_correct = false;  
 break;  
 }  
 } while(!user\_input\_is\_correct);  
 }  
}  
  
int countFalse(bool \*array, int arraySize)  
{  
 int counter = 0;  
 for (int i = 0; i < arraySize; i++)  
 if (!array[i]) counter++;  
  
 return counter;  
}

**Результат работы программы**

Работа программы начинается с меню выбора задания.

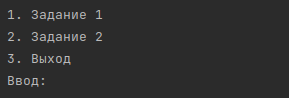


Рисунок 1. Меню выбора задания

При выборе пункта 1 появляется меню заполнения данных для локальных переменных для задания 1. После заполнения выводится информация, необходимая для вывода по заданию.

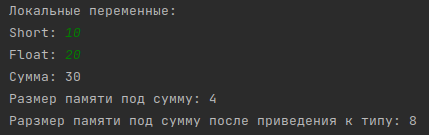


Рисунок 2. Выполнения задания 1 для локальных функций

После этого, пользователю предлагается заполнить данные для глобальных переменных.

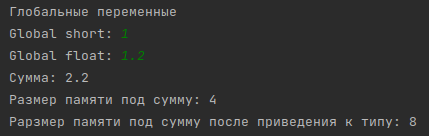


Рисунок 3. Выполнение задания 1 для глобальных функций

После выполнения задания для глобальных переменных программа возвращается к меню выбора задания(Рисунок 1).

При вводе вместо чисел некорректного значения, пользователю предлагается ввести значение еще раз.

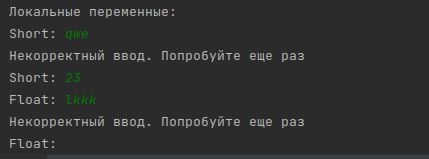


Рисунок 4. Некорректный ввод

После выбора пункта 2 в меню выбора заданий(Рисунок 1) программа запрашивает ввод значений для массива типа bool, где значение «1» соответствует значению «true», а «0» - «false». После ввода значений на экран выводится информация, необходимая для вывода по заданию.

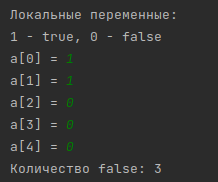


Рисунок 5. Результат работы программы для задания 2

После выполнения программы для значений локальных переменных выполняется аналогичный алгоритм для глобальных переменных.

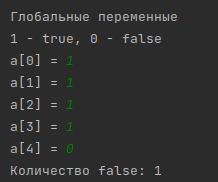


Рисунок 6. Резульатат работы программы для задания 2 для глобальных переменных

Если при вводе значений массива типа bool введено некорректное значение, пользователю будет предложено ввести значение еще раз.

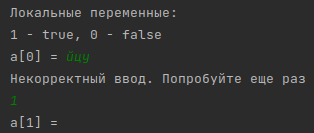


Рисунок 7. Повторный ввод значения

При выборе пункта 3 в меню выбора задания(Рисунок 1), программа завершится.

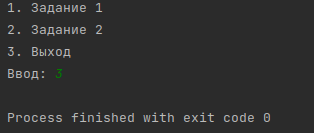


Рисунок 8. Завершение программы

**Вывод**

В результате выполнения практической работы было создано консольное приложение, позволяющее находить сумму переменных разных типов, а также подсчитывать количество элементов массива типа bool со значением «false». В процессе создания программы были использованы такие инструменты языка C++, как циклы, функции, условные операторы, глобальные переменные, а также явное приведение типов.