

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  
\_\_\_\_\_ Шибает 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:

- a) неоднократное выполнение любого задания;
- b) 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 << "Размер 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);
}

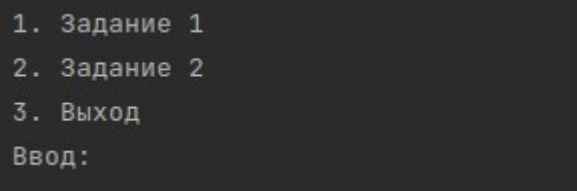
int countFalse(bool *array, int arraySize)
{
    int counter = 0;
    for (int i = 0; i < arraySize; i++)
        if (!array[i]) counter++;

    return counter;
}

```

## Результат работы программы

Работа программы начинается с меню выбора задания.



```

1. Задание 1
2. Задание 2
3. Выход
Ввод:

```

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.

```

Локальные переменные:
Short: 10
Float: 20
Сумма: 30
Размер памяти под сумму: 4
Размер памяти под сумму после приведения к типу: 8

```

Рисунок 2. Выполнения задания 1 для локальных функций

После этого, пользователю предлагается заполнить данные для глобальных переменных.

```

Глобальные переменные
Global short: 1
Global float: 1.2
Сумма: 2.2
Размер памяти под сумму: 4
Размер памяти под сумму после приведения к типу: 8

```

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.

```

Локальные переменные:
Short: qwe
Некорректный ввод. Попробуйте еще раз
Short: 23
Float: lkkk
Некорректный ввод. Попробуйте еще раз
Float:

```

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.

```

Локальные переменные:
1 - true, 0 - false
a[0] = 1
a[1] = 1
a[2] = 0
a[3] = 0
a[4] = 0
Количество false: 3

```

Рисунок 5. Результат работы программы для задания 2

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

```

Глобальные переменные
1 - true, 0 - false
a[0] = 1
a[1] = 1
a[2] = 1
a[3] = 1
a[4] = 0
Количество false: 1

```

Рисунок 6. Результат 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.

```

Локальные переменные:
1 - true, 0 - false
a[0] = 0цу
Некорректный ввод. Попробуйте еще раз
1
a[1] =

```

Рисунок 7. Повторный ввод значения

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

```
1. Задание 1
2. Задание 2
3. Выход
Ввод: 3

Process finished with exit code 0
```

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.