

Факультет Информационных систем и технологий  
Кафедра Безопасности информационных систем

**Дисциплина «Технологии программирования»**

**ОТЧЕТ**

**за практическое занятие №1**

**Тема: Управление программой при работе с простыми типами  
данных**

Выполнил  
Студент 1 курса, гр. ИСТ-262  
\_\_\_\_\_ Шибаев А. П.

Принял  
Ассистент кафедры БИС  
\_\_\_\_\_ Новиков Е.А.

**Цель работы:** получение практических навыков использования операторов цикла и условных операторов при разработке программ.

**Постановка задачи:** Разработать программу «Калькулятор», которая имитирует вычислительные действия простейшего арифметического калькулятора.

Вначале с клавиатуры вводится первое число, затем – знак арифметической операции ( +, − , \* , / ) и второе число. В зависимости от знака операции над числами выполняется соответствующее действие.

Результат операции выводится на экран и используется в следующей операции до тех пор, пока вместо знака операции не будет введен символ «С», или символ «Q». При этом, если вводится символ «С», то результат обнуляется, но калькулятор «продолжает работать», т.е. – «запрашивает» первое число и т.д.

Если вводится символ, отличный от символа знака операции, символа «С» или символа «Q», то программа «Калькулятор» продолжает работать, но «просит» повторить ввод. Если вводится символ «Q», то работа программы заканчивается с выводом последнего результата.

**Код программы:**

```
#include <iostream>
#include <limits>

template<class T> void getUserInput(T &, char const *);

int main()
{
    double number1 = 0, number2 = 0;
    char mathSign;
    bool userQuit = false;
    bool correctSignInput = true;

    std::cout << "Формат ввода: число знак число" << '\n';
    getUserInput(number1, "");
    while(!userQuit)
    {
        getUserInput(mathSign, "");
        switch (mathSign)
        {
```

```

        case 'Q':
            std::cout << number1 << '\n';
            userQuit = true;
            continue;

        case 'C':
            std::cout << "Калькулятор обнулен." << '\n';
            std::cin >> number1;
            continue;

        default:
            break;
    }
    getUserInput(number2, "");

do
{
    correctSignInput = true;
    switch (mathSign)
    {
        case '*':
            number1 *= number2;
            break;
        case '/':
            if (number2 == 0)
            {
                std::cout << "Обнаружено деление на 0.
Операция не выполняется" << '\n';
                break;
            }
            number1 /= number2;
            break;
        case '+':
            number1 += number2;
            break;
        case '-':
            number1 -= number2;
            break;
        default:
            std::cout << "Неизвестный знак. Попробуйте
еще раз." << '\n';
            std::cout << "Новый знак: "; std::cin >>
mathSign;
            correctSignInput = false;
            break;
    }
} while(!correctSignInput);

    std::cout << number1 << '\n';
}

return 0;
}

```

```

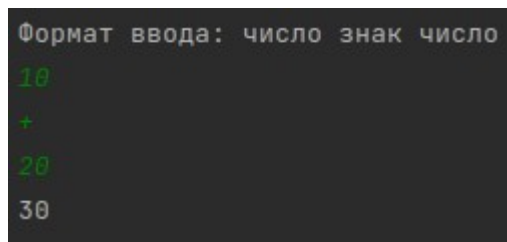
template<class T> void getUserInput(T &data, char const *
message)
{
    std::cout << message;
    while(!(std::cin >> data))
    {
        std::cout << "Некорректный ввод. Попробуйте еще раз" <<
std::endl;
        std::cin.clear();

        std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\\
n');
        std::cout << message;
    }
}

```

### Результат работы программы:

Для начала работы программы необходимо ввести число, знак операции и еще одно число.



```

Формат ввода: число знак число
10
+
20
30

```

Рисунок 1. Сумма двух чисел

Далее, необходимо вводить знак операции и еще одно число, так как в выражении будет использоваться результат предыдущей операции.

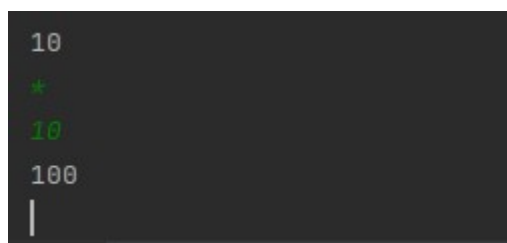


```

20
30
/
3
10

```

Рисунок 2. Деление результата на новое число



```

10
*
10
100
|

```

Рисунок 3. Умножение результата на новое число

```
100
-
99
1
```

Рисунок 4. Вычитание нового числа из результата

Если встречено деление на ноль, операция выполнена не будет.

```
1
/
0
Обнаружено деление на 0. Операция не выполняется
1
```

Рисунок 5. Деление на ноль

Если вместо допустимого математического знака введен неизвестный символ, операция выполнена не будет и программа запросит ввод нового знака.

```
1
p
10
Неизвестный знак. Попробуйте еще раз.
Новый знак: |
```

Рисунок 6. Ввод нового знака

Если вместо числа будет введено что-то, отличное от числа, операция выполнена не будет и программа запросит ввод нового числа.

```
150
c
Некорректный ввод. Попробуйте еще раз
|
```

Рисунок 7. Ввод нового числа

Если вместо знака будет введен символ «С», результат калькулятора будет обнулен и программа запросит два числа вместо одного для продолжения работы.

```
2500
C
Калькулятор обнулен.
|
```

Рисунок 8. Обнуление калькулятора

Если вместо знака будет введен символ «Q», будет выведен последний результат калькулятора и программа завершится.

```
20
Q
20
Process finished with exit code 0
```

Рисунок 9. Выход из программы

### Вывод

В результате выполнения практической работы было создано консольное приложение, позволяющее производить простейшие математические расчеты. Для написания кода программы были использованы такие средства языка C++, как условные операторы и циклы.