

# Валидация (проверка на корректность) вводимых данных

Пример простейшей программы, ожидающей ввод числа:

```
#include <iostream>
using namespace std;

void main()
{
    setlocale(0, "rus");

    int number;
    cout << "Введите число: " << endl;
    cin >> number;
    cout << number << endl;

    system("pause");
}
```

Введите число:  
123  
123  
Для продолжения нажмите любую клавишу . . .

OK

Введите число:  
123qwerty  
123  
Для продолжения нажмите любую клавишу . . .

Почти OK

Введите число:  
qwerty  
-858993460  
Для продолжения нажмите любую клавишу . . .

Неинициализированная  
переменная

# Валидация (проверка на корректность) вводимых данных

---

Улучшаем ситуацию с помощью след. методов:

**cin.good()** // возвращает true, если из потока удалось получить переменную заявленного типа (cin.fail() даст противоположный результат, т.е. true в случае возникновения ошибки)

**cin.clear()** // сбрасывает флаги ошибок, в противном случае повторный вызов cin не сработает

**cin.ignore(numeric\_limits<streamsize>::max(), '\n')** // очистка потока (отбрасываем максимальное число символов, которое может содержать поток, до первого перевода строки включительно)



# Валидация (проверка на корректность) вводимых данных

Простейшая защита от некорректного ввода:

```
int number;  
while (true)  
{  
    cout << "Введите число: " << endl;  
    cin >> number;  
    if (cin.good()){  
        break;  
    }  
    else{  
        cin.clear();  
        cin.ignore(numeric_limits<streamsize>::max(), '\n');  
        cout << "Некорректный ввод!";  
    }  
}  
cout << number;
```

Введите число:  
123  
123  
Для продолжения нажмите любую клавишу . . .

OK

Введите число:  
123qwerty  
123  
Для продолжения нажмите любую клавишу . . .

Почти OK

Введите число:  
qwerty  
Некорректный ввод!Введите число:  
qwerty123  
Некорректный ввод!Введите число:  
-45  
-45Для продолжения нажмите любую клавишу . . .

OK

# Валидация (проверка на корректность) вводимых данных

---

! Нельзя менять последовательность команд

```
cin.clear();  
cin.ignore(numeric_limits<streamsize>::max(), '\n');
```

или убирать их, иначе – бесконечный вывод строки с ошибкой и запрос на ввод без возможности произвести этот ввод

## Валидация (проверка на корректность) вводимых данных

---

Если у Вас подключена библиотека `windows.h`, то возникнет конфликтная ситуация с методом `max()`. Тогда вместо:

```
cin.ignore(numeric_limits<streamsize>::max(), '\n');
```

используйте:

```
cin.ignore(256, '\n');
```

В качестве 256 Вы можете указать и другое число. Важно понимать: это то максимально возможное количество символов, которое будет удалено из потока.

# Валидация (проверка на корректность) вводимых данных

Улучшенный вариант:

```
int number;
while (true)
{
    cout << "Введите число: " << endl;
    cin >> number;
    if (cin.get() == '\n') {
        break;
    }
    else{
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "Некорректный ввод!";
    }
}
cout << number;
```

Введите число:  
123  
123  
Для продолжения нажмите любую клавишу . . .

Введите число:  
123qwerty  
Некорректный ввод!Введите число:  
-234  
-234Для продолжения нажмите любую клавишу . . .

Введите число:  
qwerty  
Некорректный ввод!Введите число:  
qwerty123  
Некорректный ввод!Введите число:  
-45  
-45Для продолжения нажмите любую клавишу . . .



# Полный текст программы корректного ввода числа требуемого диапазона (например, для возраста из отрезка [1; 99])

```
#include <iostream>
using namespace std;

const int LEFT_RANGE_OF_AGE = 1;
const int RIGHT_RANGE_OF_AGE = 99;

int inputNumber(int left_range, int right_range);
bool isNumberNumeric();
bool isNumberRangeCorrect(int number, int left_range, int right_range);

void main()
{
    setlocale(0, "rus");

    cout << "Введите Ваш возраст: ";
    int age = inputNumber(LEFT_RANGE_OF_AGE, RIGHT_RANGE_OF_AGE);
    cout << age;

    system("pause");
}
```

Часть 1



# Полный текст программы корректного ввода числа требуемого диапазона (например, для возраста из интервала (0; 100))

## Часть 2

```
int inputNumber(int left_range, int right_range)
{
    int number;
    while (true)
    {
        cin >> number;
        if (isNumberNumeric() && isNumberRangeCorrect(number, left_range, right_range))
        {
            return number;
        }
        else
        {
            cout << "Некорректный ввод!";
            cout << "Введите корректные данные: ";
        }
    }
}
```



# Полный текст программы корректного ввода числа требуемого диапазона (например, для возраста из интервала (0; 100))

---

## Часть 3

```
bool isNumberNumeric()
{
    if (cin.get() == '\n') return true;
    else
    {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        return false;
    }
}

bool isNumberRangeCorrect(int number, int left_range, int right_range)
{
    if ((number >= left_range) && (number <= right_range)) return true;
    else return false;
}
```