**Назначение:** научится реализовывать простейший циклический алгоритм while в С++.

**Теория**

Любая реальная программа для компьютера содержит специальные операторы цикла. В программировании циклом называется многократное повторение какого-либо участка программы. Цикл в программе оформляется особым образом и компьютер понимает, что этот участок требуется повторять.

Применяется два основных вида циклов: цикл с известным числом повторений и цикл с неизвестным числом повторений.

На этом занятии рассмотрим второй вид цикла: с неизвестным числом повторения. На самом деле пользователь обычно знает количество повторений, но программа этого не «знает». Таким образом окончание цикла зависит от пользователя.

Оператор для такого вида цикла называется WHILE

Структура его следующая:

**while (условие)**

**{**

**Участок программы, который повторяется**

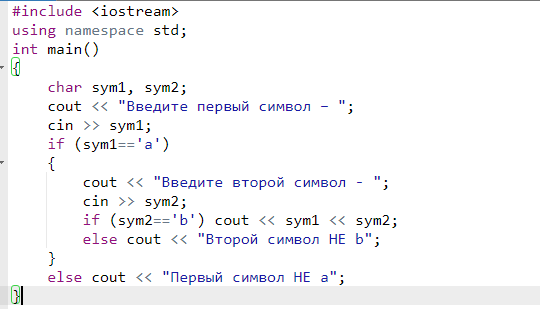
**}**

**условие** – может быть переменной типа bool, или выражением, типа p==2 (это выражение истинно, если p=2)

Цикл повторяется пока **условие** истинно и завершается, если **условие**=false

Рассмотрим два примера.

Пример 1. На предыдущем занятии был рассмотрен пример:



Для этого примера было необходимо выполнять программу 3 раза, для разных вариантов:

1. Когда все правильно
2. Когда первый символ не правильный
3. Когда первый правильно, а второй неправильно

Организуем цикл

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**char sym1, sym2;**

**while (true)**

**{**

**cout << "Введите первый символ – ";**

**cin >> sym1;**

**if (sym1=='a')**

**{**

**cout << "Введите второй символ - ";**

**cin >> sym2;**

**if (sym2=='b') cout << sym1 << sym2;**

**else cout << "Второй символ НЕ b";**

**}**

**else cout << "Первый символ НЕ a";**

**bool priznac;**

**cout << endl << "Продолжить? 1 - да, 0 - нет ";**

**cin >> priznac;**

**if (priznac) continue;**

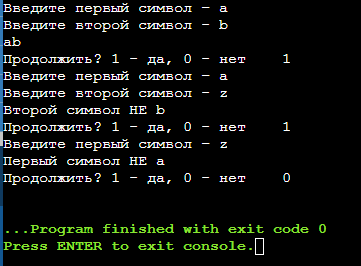
**else break;**

**}**

**}**

Оператор **break** – выход из цикла, **continue** – продолжение цикла

Синим цветом покрашен код для проверки продолжения цикла. Результат работы программы:



Второй вариант цикла для такой же программы:

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**char sym1, sym2;**

**bool priznac=true;**

**while (priznac)**

**{**

**cout << "Введите первый символ – ";**

**cin >> sym1;**

**if (sym1=='a')**

**{**

**cout << "Введите второй символ - ";**

**cin >> sym2;**

**if (sym2=='b') cout << sym1 << sym2;**

**else cout << "Второй символ НЕ b";**

**}**

**else cout << "Первый символ НЕ a";**

**cout << endl << "Продолжить? 1 - да, 0 - нет ";**

**cin >> priznac;**

**}**

**}**

Код немного сократился при этом, но программа работает аналогично. Для понимания следует вспомнить, что в с++ значение true – это 1, а false – 0.

**Важно!!!**

Рассмотрим еще один особый случай, когда вводятся по-отдельности два целых **числа**. При этом контроль ввода выполняется для каждого числа. Например,

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int a,b;**

**bool priznak=true;**

**while(priznak)**

**{**

**cout <<" введите первое целое число - ";**

**cin >> a ;**

**if (cin)**

**{**

**cout <<"ввести второе целое число - ";**

**cin >> b;**

**if (cin) cout << "Второе введено правильно";**

**else cout << "Второе - не число";**

**}**

**else cout << "Первое - не число";**

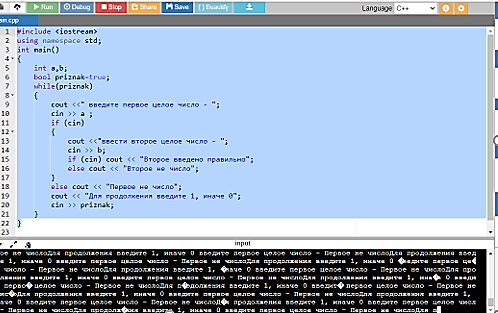
**cout << endl << "Для продолжения введите 1, иначе 0";**

**cin >> priznak;**

**}**

**}**

При запуске программа «зациклит» в том случае, если вместо первого или второго числа введен символ или знак. Вот скриншот выдачи:



Это происходит потому, что входной поток, например, cin >> a при ошибке формирует признак ошибки. Поэтому, для того что бы продолжить ввод, требуется удалить признак ошибки и очистить входной поток от старой информации. Далее приведен код с этими корректировками:

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int a,b;**

**bool priznak=true;**

**while(priznak)**

**{**

**cout <<" введите первое целое число - ";**

**cin >> a ;**

**if (cin)**

**{**

**cout <<"ввести второе целое число - ";**

**cin >> b;**

**if (cin) cout << "Второе введено правильно";**

**else cout << "Второе не число";**

**}**

**else cout << "Первое не число";**

**cin.clear(); //удаляет признак ошибки ввода**

**cin.ignore(); //чистит входной поток**

**cout << endl << "Для продолжения введите 1, иначе 0";**

**cin >> priznak;**

**}**

**}**

**Практика**

Свою программу из задания 6 дополнить циклом while и проверить все варианты за один запуск.