СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФМИ, Софтуерно инженерство

ОПЕРАТОРИ ЗА ЦИКЪЛ [FOR],[WHILE], [DO/WHILE]

Въведение

•Защо са ни необходими?

Случвало ли ви се е да ви накажат да напишете „Няма да говоря в час“ 1000 пъти?

Колко по-лесно е веднъж да напишете:

```
for (unsigned i = 0; i < 1000; ++i)
        std::cout<<“Няма да говоря в час\n”;
```

Q:Какво е цикъл?
A: Процес, който представлява многократното изпълнение на дадена последователност от операции с еднакви/различни данни.
A:(неформално): Цикъл е операция, която се повтаря N на брой пъти

Q:Защо да използваме цикъл?
A:Спестява ръчното писане на един и същ код

Въведение

- **ЦИКЛИЧЕН ИЗЧИСЛИТЕЛЕН ПРОЦЕС**

Изчислителен процес, при който оператор или група оператори се изпълняват многократно за различни стойности на техните параметри, се нарича **цикличен.**

Индуктивен цикличен процес – цикличен процес, при който **броят на повторенията е предварително известен**

Итеративен цикличен процес – цикличен процес, при който **броят на повторенията НЕ е известен предварително.**

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФМИ, Софтуерно инженерство

УСЛОВЕН ОПЕРАТОР WHILE

- **СЕМАНТИКА НА ОПЕРАТОРА WHILE**

Чрез този оператор може да се реализира **произволен цикличен процес.**

```
while(<условие>)
{
  <действия> //Тяло на цикъла
}
```

Пример:

```
int i = 1; //инициализация
Int a = 5; //инициализация

while(i <= a) //условие
{
    //тяло на цикъла
    cout << i+a << endl;
    i++; //актуализация
}
```

Докато е в сила условието, повтаряй следното...

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФМИ, Софтуерно инженерство

УСЛОВЕН ОПЕРАТОР WHILE

- **СЕМАНТИКА НА ОПЕРАТОРА WHILE**

while (докато) – запазена дума
<условие> - булев израз
<действия> – инициализира се кодът, който трябва да се изпълни определен брой пъти

Пресмята се стойността на <условие>.
Ако тя е false изпълнението на оператора while завършва без да се изпълни тялото му.
Ако тя е true, изчисляването на условието и изпълнението на **<действия>** продължава, докато <условие> = **false**

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФМИ, Софтуерно инженерство

УСЛОВЕН ОПЕРАТОР WHILE

- **Примери**
  - **Пример 1:**

```
unsigned i = 0;
while(i < 1000) {
        std::cout << i; //ще изпише числата от 0 до 999
        ++i;
}
```

  - **Пример 2: Безкраен цикъл**

```
while (true)
        std::cout << “I love C++\n”; //безкраен цикъл
```

  - **Пример 3: Break/Безкраен цикъл с break**

```
unsigned i = 0;
while(true)
{
        if(i == 99)
                break; //ето как можем да счупи безкраен цикъл
                       //след 100 стъпки
        else
                ++i;
}
```

УСЛОВЕН ОПЕРАТОР WHILE

- **Примери**
  - **Пример 4:** Принтиране на всички четни числа от 1 до 1000

```
int counter = 0;
while (counter < 1000)
{
      ++counter;
      if (counter % 2 != 0)
            continue;
      std::cout << counter << “ “;
}
```

- **Пояснение:**

1. Ако break се извика в цикъл, цикълът се прекъсва
2. Оператор continue прекратява командата в цикъла
3. За разлика от оператор break, оператор continue не прекратява цикъла, а само командата, като цикълът продължава все едно нищо не се е случило

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФМИ, Софтуерно инженерство

УСЛОВЕН ОПЕРАТОР DO/WHILE

- **СЕМАНТИКА НА ОПЕРАТОРА DO/WHILE**

```
do
{
    <действия> //Тяло на цикъла
}
while(<условие>);
```

```
int main()
{
    int i, a;
    i = 1; //инициализация
    a = 0; //инициализация
    do
    {
        //<действия>
        cout << i+a << endl;
        i++; //актуализация
    }
    while(i <= a); //условие
}
```

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФМИ, Софтуерно инженерство

УСЛОВЕН ОПЕРАТОР DO/WHILE

• СЕМАНТИКА НА ОПЕРАТОРА DO/WHILE

do (направи/прави), while (докато) – запазени думи
<действия> - инициализира се кодът, който трябва да се изпълни определен брой пъти
Важно: Тялото се изпълнява поне веднъж.
<условие> - булев израз

Изпълнява се тялото на цикъла **поне веднъж**, след което се пресмята стойността на **<условие>.**
Ако то е **false,** изпълнението на оператора do/while **завършва**.
В противен случай се повтарят действията:

- Изпълнение на тялото на цикъла
- Пресмятане на стойността на **<условие>,** докато **<условие>** = true.

УСЛОВЕН ОПЕРАТОР DO-WHILE

- **Примери**
  - **Пример 1: Hello World**

```
int main()
{
   int i = 2;
   do
   {
        cout << "Hello World\n";
        i++;
   }while (i < 1);
}
```

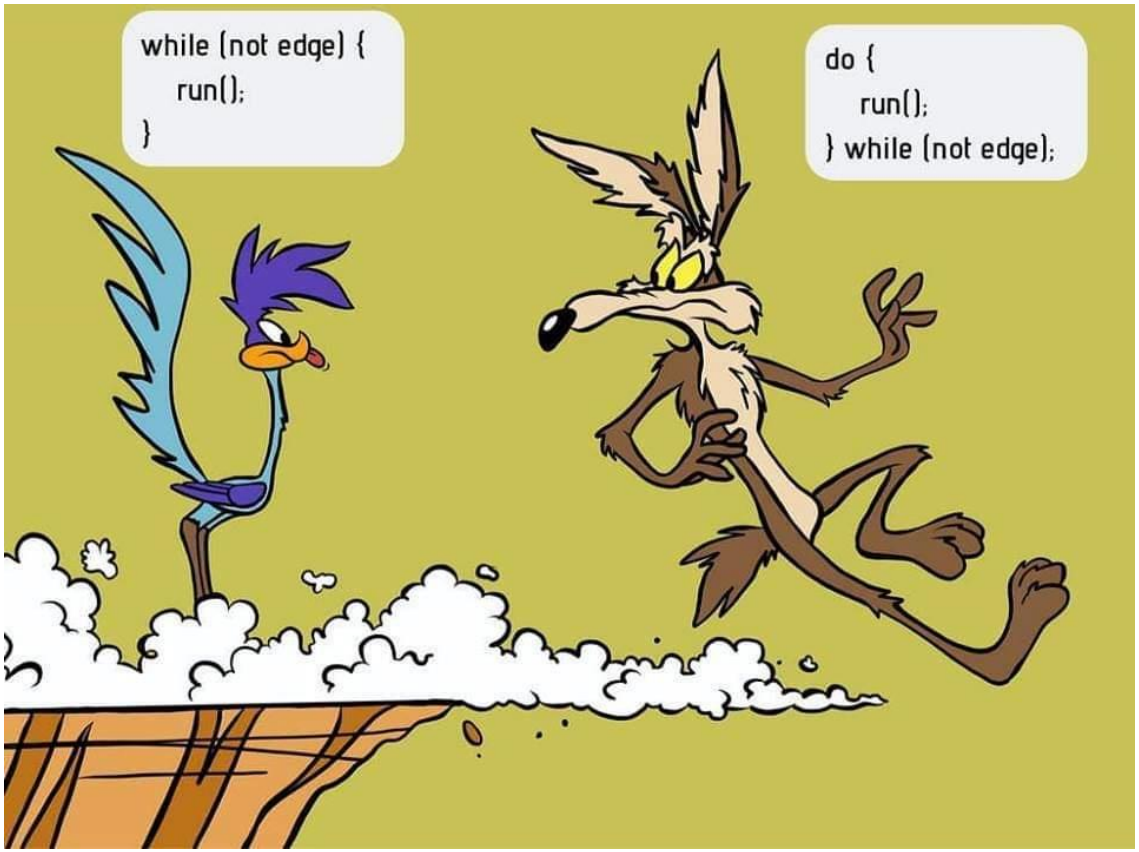
Пример 2:

```
int a = 0;
do
{
     cout << "value of a: " << a << endl;
     a += 1;
}while( a < 20 );
```

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФМИ, Софтуерно инженерство

WHILE vs DO-WHILE



СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФМИ, Софтуерно инженерство

ОПЕРАТОР ЗА ЦИКЪЛ FOR

- **СЕМАНТИКА НА ОПЕРАТОРА FOR**

```
for ( <инициализация>; <условие> ; <корекция> )
{
    <действия> //Тяло на цикъла
}

int main()
{
   int n;
   n = 5;

   for(int i = 1; I <= 5; i++)
   {
      cout << n * i << endl;
   }
}
```

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФМИ, Софтуерно инженерство

ОПЕРАТОР ЗА ЦИКЪЛ FOR

- **СЕМАНТИКА НА ОПЕРАТОРА FOR**

```
for ( <инициализация>; <условие> ; <корекция> )
{
    <действия> //Тяло на цикъла
}
```

for - запазена дума
<инициализация> - задава начало на изпълнението, като тя се изпълнява **само веднъж в началото на цикъла.**
<условие> - булеви израз. Ако стойността му е **false** изпълнението на for завършва **без тялото на цикъла да се изпълни.**
Ако стойността му е **true се повтарят следните действия:**
- изпълнение на тялото на цикъла
- корекция/актуализация
- пресмятане на стойността на условие, **докато е true**

Примерни задача

- **Задача естествени числа**

Да се състави програма, в която потребителят въвежда естествено число, а програмата извежда неговите делители.

Примерни задача

- **Решение, чрез FOR цикъл**

```
int main()
{
   int a, n;
   cout << "Please, enter a number" << endl;
   cin >> a;

   for( int i=1 ; i <= a/2 ; i++)
   {
      n = a % i;
      if(n==0)
      {
         cout << i << endl;
      }
   }
   cout << a << endl;
   return 0;
}
```

Примерни задача

- **Решение, чрез WHILE цикъл**

```
int main()
{
    int a, n, i;
    i = 1;
    cout << "Please, enter a number" << endl;
    cin >> a;
    while(i<=a)
    {
        n = a % i;
        if(n==0)
        {
            cout << i << endl;
        }
        i++;
    }
    return 0;
}
```

Примерни задачи

- **Решение, чрез DO/WHILE цикъл**

```
int main()
{
   int a, n, i = 1;
   cout << "Please, enter a number" << endl;
   cin >> a;
   do
   {
      n = a % i;
      if(n==0)
      {
         cout << i << endl;
      }
      i++;
   }
   while(i <= a);
   return 0;
}
```

Вложени цикли

- **Пример:**

Да се изведат всички числа от 1 до 100 по хиляда пъти.

```
for(unsigned i = 1; i <= 100; ++i)
{
    for(unsigned j = 0; j < 1000; j++)
    {
        std::cout << i << " ";
    }
}
```

Примерни задачи

- **GCD – Greatest common divisor**

Да се състави програма, в която потребителят въвежда 2 естествени числа, а програмата извежда техния НОД.

- **Reverse and increment**

Да се състави програма, в която потребителят въвежда естествено число, като програмата обръща реда на цифрите му(123 -> 321) и го инкрементира.

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФМИ, Софтуерно инженерство

ВЪПРОСИ