



قسم الهندسة المعلوماتية
مقرر مدخل إلى الخوارزميات والبرمجة

محاضرات الأسبوع 8 الفصل الثاني

2023/2022

التعريف بشكل تعليمة حلقة while في لغة ++c

يمكن التعبير عن الشكل العام لهذه البنية كما يلي:

```
while (boolean_expression)  
{ statements }
```

التنفيذ يكون وفق الآلية والترتيب التالي:

١. يتم اختبار الشرط
٢. إذا كان ناتج تقييم الشرط true يتم تنفيذ statements والتي يجب أن تحوي أمر يغير بقيمة الشرط المنطقية.
٣. العودة إلى الخطوة رقم ٢.

يستمر تكرار الحلقة إلى أن يصبح ناتج تقييم الشرط false، تتوقف الحلقة ويتم الانتقال لتنفيذ الأمر الذي يلي حلقة while.

حيث:

- while هي كلمة مفتاحية.
- boolean_expression هي تعبير بلغة ++C.
- statement أمر بسيط أو مجموعة أوامر.

حلقة do-while في لغة ++c

التنفيذ يكون وفق الآلية والترتيب التالي:

١. يتم تنفيذ statement.
٢. يتم اختبار الشرط
٣. إذا كان ناتج تقييم الشرط true يعود لتنفيذ statement. والتي يجب أن تحوي أمر يغير بقيمة الشرط المنطقية.

يستمر التكرار إلى أن يصبح ناتج تقييم الشرط false، تتوقف الحلقة ويتم الانتقال لتنفيذ الأمر الذي يلي الحلقة. لكن مع الانتباه أنها تنفذ لمرة واحدة قبل التحقق من الشرط مما يسبب أخطاء بحالات خاصة.

يمكن التعبير عن الشكل العام لهذه البنية كما يلي:

```
do  
{ statements;}  
while (boolean_expression);
```

حيث:

- Do-while هي كلمة مفتاحية.
- boolean_expression هي تعبير بلغة ++C.
- statements أمر بسيط أو مجموعة أوامر.

حلقة for في لغة ++c

يمكن التعبير عن الشكل العام لهذه البنية كما يلي:

```
for ( init_expression; boolean_expression; step_expression )  
    { statement }
```

التنفيذ يكون وفق الآلية والترتيب التالي:

١. يتم تنفيذ أمر التهيئة `initialization_expression`.
 ٢. يتم اختبار الشرط.
 ٣. إذا كان ناتج تقييم الشرط `true` يتم تنفيذ `statement`.
 ٤. يتم تنفيذ الخطوة `step_expression`.
 ٥. العودة إلى الخطوة رقم ٢.
- يستمر التكرار إلى أن يصبح ناتج تقييم الشرط `false`،
تتوقف الحلقة ويتم الانتقال لتنفيذ الأمر التالي للحلقة `for`.

حيث:

- `for` هي كلمة مفتاحية.
- `init_expression, boolean_expression,`
- `step_expression` تعابير ++C.
- `statement` أمر بسيط أو مجموعة أوامر.

برنامج لإيجاد ناتج قسمة عددين صحيحين دون استخدام
معامل القسمة حيث x, y عددين موجبين

```
#include <iostream>
using namespace std;
int main()
{ int x,y,i=0 ;
cout<<"enter 2 int: ";
cin>>x >>y;
while(x >=y)
{x=x-y ; i++ ;}
cout<<"x /y= "<<i<<endl;
return 0;
}
```

أدخل عددين $x=5$ و $y=8$ وبرر الخرج

```
#include <iostream>
using namespace std;
int main()
{ int x,y,i=0 ;
cout<<"enter 2 int: ";
cin>>x >>y;
do
{x=x-y ; i++ ;}
while(x >=y);
cout<<"x / y="<<i<<endl;
return 0;
}
```

أدخل عددين $x=5$ و $y=8$ وبرر الخرج

```
#include <iostream>
using namespace std;
int main()
{ int x,y,i ;
cout<<"enter 2 int: ";
cin>>x >>y;
for( i=0; x >=y;i++ ) {x=x-y ;}
cout<<"x / y="<<i<<endl;
return 0;
}
```

أدخل عددين $x=5$ و $y=8$ وبرر الخرج

```
#include <iostream>
using namespace std;
int main()
{ int counter,total,average, grade;
total = 0;counter = 1;
while (counter <= 10)
{
cout << "Enter grade: ";
cin >> grade;
total = total + grade;
counter = counter + 1;
}
average = total / 10;
cout << "Class average is ";
cout<< average << endl;
return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{ float average; int counter,grade,total;
total=0;counter=0;
cout << "Enter grade (0..100) out of range to end:";
cin >> grade;
while ( (grade >= 0) && (grade <= 100) )
{total = total + grade; counter = counter + 1;
cout<<" Enter grade (0..100) out of range to end: ";
cin >> grade; }
if (counter != 0)
{ average = (float) total / counter;
cout<<"Class average is"<<average; }
else cout<<"No grades were entered \n";
return 0;
}
```

تتبع البرنامجين التاليين
وحدد الفرق بينهما من
خلال رسم مخطط
تدفقي لكل منهما :
نلاحظ في البرنامج الأول
يوجد عدد محدد من
التكرارات وهو ١٠
أما في البرنامج الثاني
فعدد التكرارات غير
محدد.

انتهت تمارين الأسبوع ٨

جامعة
المنارة
HIMARA UNIVERSITY