****

**Programmation Java**

**S05 – Boucles for/while/do..while**

**Claudio Leitao, Luke Perrottet**

**lundi, 2 octobre 2017**

# Exercice 1

1. Première exécution : sum vaut 5

Deuxième exécution : sum vaut 9

Troisième exécution : sum vaut 12

Quatrième exécution : sum vaut 14

Cinquième exécution : sum vaut 15

La valeur affichée sera : 15

1. La boucle while sera effectuée 5 fois mais sum ne sera modifié qu’à la fin de ces 5 exécutions. Donc sum vaudra 0 car n vaudra 0 à la fin de la boucle while.

La boucle for sera exécuté 4 fois :

Première exécution : sum vaut 1

Deuxième exécution : sum vaut 3

Troisième exécution : sum vaut 7

Quatrième exécution : sum vaut 15

La valeur affichée sera 15

1. Première exécution : sum vaut 4

Deuxième exécution : sum vaut 7

Troisième exécution : sum vaut 9

Quatrième exécution : sum vaut 10

Cinquième exécution : sum vaut 10 (car n = 0)

La valeur affichée sera : 10

1. Boucle infinie car la boucle while est toujours vraie. La variable n n’est jamais modifiée.
2. À la première itération, sum vaudra 16 car la deuxième boucle sera exécutée 4 fois.

À la deuxième itération, n vaut 3 donc la deuxième boucle est ignorée car le continue fait foi.

À la troisième itération, sum vaut 20 car la deuxième boucle sera exécutée 2 fois.

À la quatrième itération, sum vaut 21 car la deuxième boucle sera exécutée 1 fois.

À la fin sum vaut 23 car la dernière boucle ne sera exécutée que deux fois, vu que m vaut 3 à la fin de la première boucle.

1. Boucle infinie car la représentation d’un double est une approximation ( 1 puissance –n) et double n’est capable que d’afficher 16 chiffres significatifs après la virgule.

# Exercice 2

**package** s05;

**public** **class** **Ex02** {

**public** **static** String **dentScie**(**int** ligne, **int** colonne){

String message = "";

**for** (**int** i = **1**; i <= ligne ; i++) {

**for** (**int** j = **1**; j <= colonne; j++) {

**for** (**int** k = **1**; k <= j; k++) {

message += "\*";

}

message += "\n";

}

}

**return** message;

}

**public** **static** **void** **main**(String args[]) {

System.out.println(dentScie(**3**, **4**));

}

}

## Résultat

\*

\*\*

\*\*\*

\*\*\*\*

\*

\*\*

\*\*\*

\*\*\*\*

\*

\*\*

\*\*\*

\*\*\*\*

# Exercice 3

**package** s05;

**public** **class** **Ex03** {

**public** **static** **void** **main**(String args[]) {

**double** e = **1.0**;

**double** nFact = **1.0**;

**double** addValue = **1.0**;

**double** i = **1.0**;

**while** (addValue > **1.0**E-**12**) {

e += addValue;

nFact \*= ++i;

addValue = **1.0** / nFact;

System.out.println("e " + e);

}

}

}

## Questions

Combien d’itérations sont nécessaire ?

14 itérations

Pourrait-on se contenter du type float ?

Oui, « e » serait sur 6 chiffres après la virgule

|  |  |
| --- | --- |
| **Double** | **Float** |
| e = 2.0  e = 2.5  e = 2.6666666666666665  e = 2.708333333333333  e = 2.7166666666666663  e = 2.7180555555555554  e = 2.7182539682539684  e = 2.71827876984127  e = 2.7182815255731922  e = 2.7182818011463845  e = 2.718281826198493  e = 2.7182818282861687  e = 2.7182818284467594  e = 2.71828182845823 | e = 2.0  e = 2.5  e = 2.6666667  e = 2.7083335  e = 2.716667  e = 2.7180557  e = 2.718254  e = 2.718279  e = 2.7182817  e = 2.718282  e = 2.718282  e = 2.718282  e = 2.718282  e = 2.718282 |

# Exercice 4

**package** s05;

**public** **class** **Ex04** {

**public** **static** String **firstFact**(**int** num) {

**int** div = **2**;

String message = num + " = ";

**while** (num != **1**) {

**if** (num % div == **0**) {

num /= div;

message += div + " x ";

} **else** {

**if** (div == **2**)

div++;

**else**

div += **2**;

}

}

**return** message.substring(**0**, message.length() - **2**);

}

**public** **static** **void** **main**(String args[]) {

System.out.println(firstFact(**120**));

}

}

## Résultat

**120** = **2** x **2** x **2** x **3** x **5**

# Exercice 5

**package** s05;

**public** **class** **Ex05** {

**public** **static** **void** **main**(String args[]) {

String message = "";

**int** i = **50**;

**int** numInLine = **0**;

**while** (i <= **1000**) {

**if** (i % **5** == **0** && i % **6** == **0**) {

**if** (++numInLine < **10**) {

message += i + ",";

} **else** {

numInLine = **0**;

message += i + "\n";

}

}

i++;

}

System.out.println(message.substring(**0**, message.length() - **1**));

}

}

## Résultat

**60**,**90**,**120**,**150**,**180**,**210**,**240**,**270**,**300**,**330**

**360**,**390**,**420**,**450**,**480**,**510**,**540**,**570**,**600**,**630**

**660**,**690**,**720**,**750**,**780**,**810**,**840**,**870**,**900**,**930**

**960**,**990**