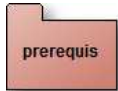


NFP121 : Programmation avancée
TP n° 2 - Conversion Fahrenheit - Celsius



Lectures préalables :

- Java_I
- [The Java™ Tutorials](#) (*Getting Started et Learning the Java Language*)

Thèmes du TP :

- Utilisation de [BlueJ](#)
- Java impératif
- Exceptions

Ce TP est consacré à une application Java de conversion des degrés Fahrenheit (Angleterre) en degrés Celsius (reste de l'Europe!).



La classe "FahrenheitCelsius"

Ouvrir avec Bluej le projet "tp2", descendre dans le paquetage "question1" où l'on découvre la classe "FahrenheitCelsius".

C'est dans cette classe qu'est effectuée la conversion. Cette classe est une application car elle contient la méthode :

```
public static float fahrenheitEnCelsius(int f) { //..
```

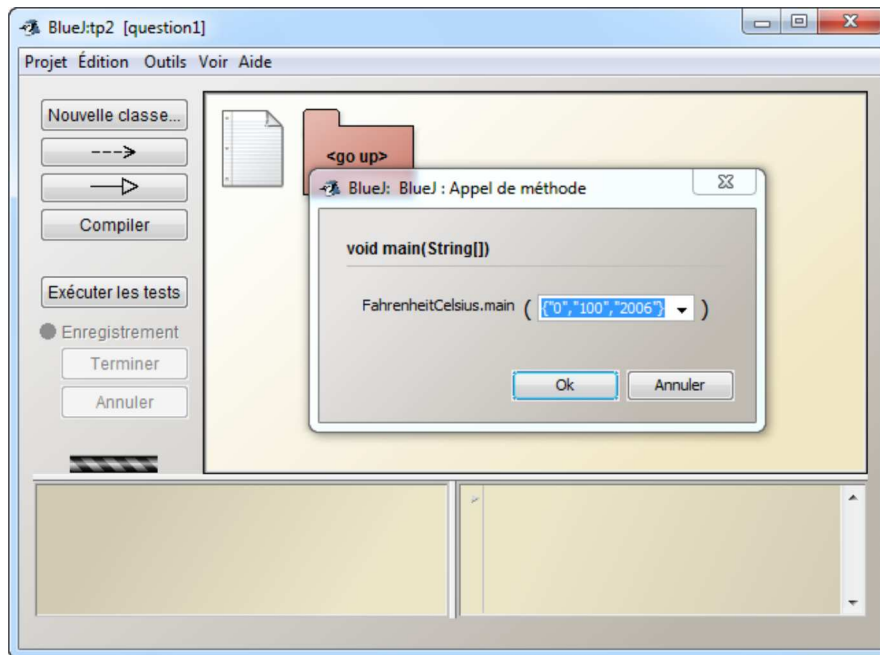
Compléter les 2 méthodes de cette classe

Contraintes :

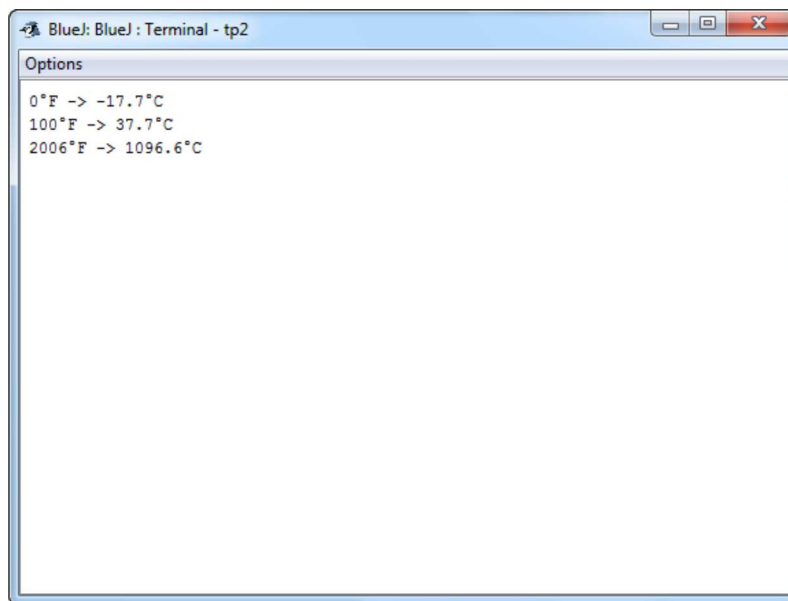
- La formule de conversion : $^{\circ}\text{C} = 5/9 * (^{\circ}\text{F} - 32)$
- *Les valeurs en degrés Fahrenheit à convertir sont données sur la ligne de commande.*
- *Ce sont des valeurs entières.*
- *Le résultat est attendu avec un seul chiffre derrière la virgule, obtenu uniquement par le calcul.* (par exemple r = ((int)resultat*10)/10.0)

Exécution sous Bluej :

"clic droit" sur le diagramme de la classe "FahrenheitCelsius" , choisir l'item "void main(String[] args)", introduire des données comme ci dessous par exemple ...

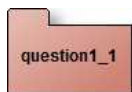


Le résultat dans la fenêtre Terminal



Technique Java :

- Les arguments de la ligne de commande sont des chaînes (String), donc pour effectuer la conversion il faut les convertir en entiers (cf. `static int parseInt(String s)` de la classe [Integer](#) du package `java.lang`)
- Pour les tableaux, le champ `length` contient le nombre d'éléments du tableau (ATTENTION le premier indice d'un tableau est 0).
- Attention : la division entre 2 entiers est une division entière . Exemple $5/9 = 0$ et $9/5 = 1$



La classe de tests unitaires "FahrenheitCelsiusTest"

Enrichissez la classe de tests unitaires comparant des résultats calculés à des résultats attendus.

Lire le tutoriel "[Unit Testing in BlueJ](#)" sur les classes de tests sous BlueJ et un exemple est donné dans le projet...

question2

La gestion de l'exception "NumberFormatException"

"runtime" Exceptions :

Descendre maintenant dans le paquetage "question2" où l'on découvre une autre version de la classe "FahrenheitCelsius".

Développez une nouvelle version de cette classe en reprenant la conversion de la question 1 et en ajoutant la gestion de l'exception "[NumberFormatException](#)".

L'exception "**NumberFormatException**" est levée lors d'une mauvaise entrée sur la ligne de commande i.e. d'une entrée autre qu'une chaîne convertible en "int" Exemple : {"10","ZZ","100"} engendre :

For input string: "ZZ" est généré par l'appel de `e.getMessage()` . e étant le nom de l'Exception soit **`System.out.println("error : " + e.getMessage());`**

question2_1

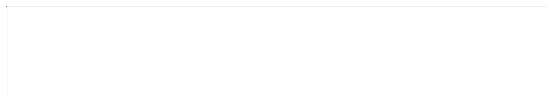
La classe de tests unitaires

Développez une classe de tests unitaires (cf. question1)

question3

Applette (applet ?)

Descendre maintenant dans le paquetage "question2" et compléter l'"Applette" fournie en utilisant la fonction de conversion de la question 2.



L'applette complétée

question3_1

Exécution sous BlueJ

- Clic droit sur le diagramme de la classe "AppletteFahrenheit" et choisir l'item "Exécuter l'applet",
- Cocher "Utiliser le visualiseur d'applets",
- Puis "ok"

question3_2

Prise en compte de l'exception "NumberFormatException"

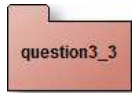
Ajoutez à cette Applette la prise en compte de l'exception "NumberFormatException" : si cette exception est levée la sortie mentionnera le libellé "error !", comme le montre cette copie d'écran



question3_3

Prise en compte du zéro absolu

Complétez l'Applette de 3.2 par la prise en compte du zéro absolu : la valeur minimale ne peut être inférieure en à -273.1°C , un fonctionnement analogue à l'applette de démonstration ($\text{if}(\text{celsius} < -273.1\text{F}) \text{celsius} = -273.1\text{F}$)

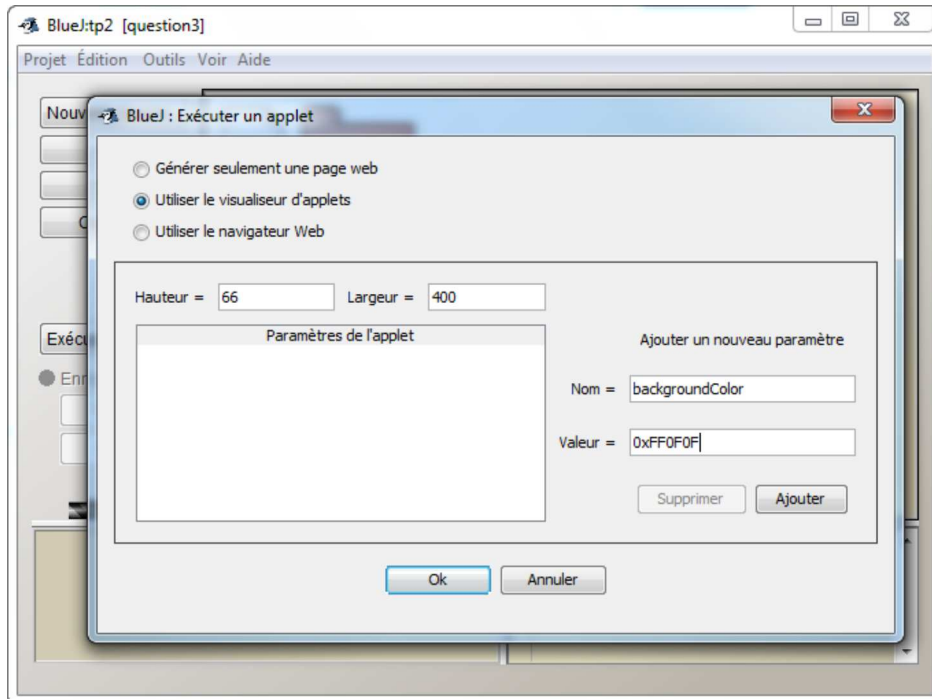


Prise en compte du paramètre "backgroundColor"

Ajoutez à cette Applette la prise en compte du paramètre "backgroundColor" (la couleur du fond).

Remarque :

- Sous BlueJ, il vous suffit d'ajouter ce paramètre comme le suggère l'interface ci-dessous et de modifier votre programme en conséquence.
- Exécuter l'applet, puis ajouter un paramètre
- Bien entendu votre programme doit continuer de fonctionner **avec ou sans** ce paramètre



Technique Java :

- La méthode "[getParameter](#)" permet de lire le paramètre "backgroundColor",
- C'est une méthode de classe : "`static Color decode(String nm);`" de la classe [java.awt.Color](#) qui doit être exécutée,
- Si une erreur se produit, c'est la couleur rose ("`Color.pink`") qui est conservée.