Leïla FALLOURD;Victor LAURENT

leila.fallour@etu.univ-lyon1.fr ; victor.laurent@etu.univ-lyon1.fr

TD 01 : les bases de Java

Table des matières

[EXERCICE 1 : 2](#_Toc471291993)

[Déclaration des variables : 2](#_Toc471291994)

[Type de conversion implicite, type et valeur des expression arithmétique suivantes : 2](#_Toc471291995)

[Seconde déclaration 3](#_Toc471291996)

[Type de conversion implicite, type et valeur des expression arithmétique suivantes (suite) 3](#_Toc471291997)

[EXERCICE 2 4](#_Toc471291998)

# EXERCICE 1 :

## Déclaration des variables :

* byte b1=10, b2=20 ;
* short p= 200 ;
* int n = 500 ;
* long q = 100 ;
* float x = 2.5f ;
* double y = 5.25 ;

## Type de conversion implicite, type et valeur des expression arithmétique suivantes :

* b1+b2 //b1 et b2 deviennent des int par promotion numérique, le résultat est un int = 30
* p+b1 // p devient un int par promotion numérique, b1 une promotion numérique, le résultat est un int = 210
* b1\*b2 // b1 et b2 deviennent des int par promotion numérique, le résultat est 200
* q+p\*(b1+b2) // b1 et b2 deviennent des int par promotion numérique puis des long par conversion d’ajustement, p devient un int par promotion numérique puis devient un long par conversion d’ajustements. Le résultat est un long = 6100
* x+q\*n // n et q deviennent des float par conversion d’ajustement, le résultat est un float = 50002.5
* b1\*q/x // b1 devient un int par promotion numérique puis un float par conversion d’ajustement, q devient un long par conversion d’ajustement. Le résultat est un float = 400.0
* b1\*q\*2./x // b1 devient un int par promotion numérique puis un double par conversion d’ajustement, q devient un double par conversion d’ajustement, 2. est un double, x devient un double par conversion d’ajustement. Le résultat est un double = 800.0
* b1\*q\*2.f/x // b1 devient un int par promotion numérique puis un float par conversion d’ajustement, q devient un float par conversion d’ajustement, 2.f est un float. Le résultat est un float = 800.0

/\* On doit faire dans l’ordre des opérations et que les byte, char et short ne passe par des int quand un opérande est muni d’opérateur \*/

## Seconde déclaration

* char c = 60, ce = ’e’, cg = ‘g’ ;
* byte b = 10 ;

## Type de conversion implicite, type et valeur des expression arithmétique suivantes (suite)

* c+1 // c devient un int par promotion numérique, 1 est déjà un int, le résultat est un int = 61
* 2\*c // c devient un int par promotion numérique, 2 est déjà un int, le résultat est un int = 120
* cg – ce // cg et ce deviennent des int par promotion numérique, le résultat est un int = 2
* b \* c // b et c deviennent des int par promotion numérique, le résultat est un int = 600

# EXERCICE 2

Après exécution, on obtient l’affichage suivant :

# EXERCICE 3

* int i=10, j=5 ;
* if(i<5 && j++<10) … ;
* //