



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

Preguntas de análisis Ejercicio 2:

A. Según el siguiente código, indique qué se imprime por consola y explique el porqué de cada línea donde se imprime.

Se imprime lo siguiente:

char: Entra a int: 103

//no hay un método específico para char, entonces entra al método que reciba como argumento un tipo de dato mayor a char en la escala de tipos primitivos pero que sea el mínimo de los posibles, en este caso int. Al entrar a un método de tipo entero, pero al ser un char almacena datos alfanuméricos, el compilador lo convierte según el código ASCII, donde "g" equivale a 103.

short: Entra a int: 2

//no hay un método específico para short, entonces entra al método que reciba como argumento un tipo de dato mayor a char en la escala de tipos primitivos pero que sea el mínimo de los posibles, en este caso int. Como short está dentro del rango que trabaja int, su valor no cambia.

byte: Entra a int: 1

//no hay un método específico para byte, entonces entra al método que reciba como argumento un tipo de dato mayor a char en la escala de tipos primitivos pero que sea el mínimo de los posibles, en este caso int. Como byte está dentro del rango que trabaja int, su valor no cambia.

long: Entra a double: 9.99999999E8

//no hay un método específico para long, entonces entra al método que reciba como argumento un tipo de dato mayor a char en la escala de tipos primitivos pero que sea el mínimo de los posibles, en este caso double porque int es menor a long en la escala. Al tratar de poner un número tan grande como el long a un double, debemos convertirlo a un decimal que valga lo mismo, pero que entre en el rango de double, usando la notación científica (9.99999999E8), donde E8 es como decir x10 a la 8.

integer: Entra a int: 51232

// Cómo si hay un método específico para int, entonces entra al método que reciba como argumentos variables de tipo entero directamente.

double: Entra a double: 12.4

// Cómo si hay un método específico para double, entonces entra al método que reciba como argumentos variables de tipo doble directamente.



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

float: Entra a double: 5.650000095367432

//no hay un método específico para float, entonces entra al método que reciba como argumento un tipo de dato mayor a char en la escala de tipos primitivos pero que sea el mínimo de los posibles, en este caso double porque int es menor a float en la escala. Se agregan más números porque al pasar de float a double se tiene más precisión.

B. Realice los siguientes cambios, teniendo siempre como referencia el código inicial.

- **Active la función que recibe un short**

char: Entra a int: 103

short: Entra a short: 2

byte: Entra a short: 1

long: Entra a double: 9.99999999E8

integer: Entra a int: 51232

double: Entra a double: 12.4

float: Entra a double: 5.650000095367432

//Lo único que **cambia** es **lo que imprime para los valores de tipo short y byte**. Short porque al tener un método que reciba exactamente su mismo tipo de dato se va a ejecutar directamente ahí y Byte porque al no tener un método que reciba exactamente su mismo tipo de dato busca el inmediatamente mayor que en este caso sería short.

- **Active la función que recibe un float**

Teniendo como referencia el código inicial:

char: Entra a int: 103

short: Entra a int: 2

byte: Entra a int: 1

long: Entra a float: 1.0E9

integer: Entra a int: 51232

double: Entra a double: 12.4

float: Entra a float: 5.65



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

Teniendo como referencia el código anterior:

char: Entra a int: 103
short: Entra a short: 2
byte: Entra a short: 1
long: Entra a float: 1.0E9
integer: Entra a int: 51232
double: Entra a double: 12.4
float: Entra a float: 5.65

//En cualquier caso, lo único que **cambia** es **lo que imprime para los valores de tipo long y float**. Float porque al tener un método que reciba exactamente su mismo tipo de dato se va a ejecutar directamente ahí y queda el mismo valor pues es el mismo tipo de dato y long porque al no tener un método que reciba exactamente su mismo tipo de dato busca el inmediatamente mayor que en este caso sería float. El long al entrar al float se convierte en decimal, expresándolo en notación científica, pero como el float no tiene tanta precisión como el long, lo termina aproximando un poco pues son muchos decimales.

- **Comente la función que recibe un double y active la que recibe un float.**

En este caso imprimiría **error** porque **double sólo puede entrar a los métodos que reciban su mismo tipo** y en teoría uno mayor, pero **no hay tipos de datos primitivos mayores a double** y si se comenta la línea que recibe double, este valor no podría ser pasado a ningún método.

- **Comente todas las funciones, excepto la que recibe un doble.**

char: Entra a **double: 103.0**
short: Entra a **double: 2.0**
byte: Entra a **double: 1.0**
long: Entra a **double: 9.999999999E8**
integer: Entra a **double: 51232.0**
double: Entra a **double: 12.4**
float: Entra a **double: 5.650000095367432**

En este caso, **todos entrarían al método que recibe double** porque es el único disponible y, además, **recibe al tipo de dato más grade que hay en java**, por lo que será mayor a todos y por tanto **pueden ser pasados como argumento**. A los valores que caben tal como están en el double solo les agrega (.0), al resto, los expresa en notación científica (un decimal por 10 que sería E, elevado a un número).