



Flujo, ciclos y métodos

Casting en Java

***Reconocer las
características
fundamentales del lenguaje
Java para el desarrollo de
aplicaciones empresariales***

- Unidad 1: Flujo, ciclos y métodos
- Unidad 2: Arreglos y archivos
- Unidad 3: Programación orientada a objetos
- Unidad 4: Pruebas unitarias y TDD



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Identificar los tipos de datos de una variable para comprender formas de transformar desde un tipo de dato a otro.*
- *Aplicar transformaciones de tipos de datos de una variable.*

¿Qué son los operadores aritméticos?



`/* Casting */`

Implementación de un casting

Casting: procedimiento para transformar una variable primitiva de un tipo a otro, o transformar un objeto de una clase a otra clase siempre y cuando tengan relación entre sí.

¿Cómo se realiza?

Se coloca el tipo de dato al cual queremos convertir antecediendo al elemento que deseamos su transformación

```
System.out.println((int) 4.5); // 4  
System.out.println((float) 4); // 4.0
```

Casting en String

Dato alfanumérico: podemos inferir que un String puede entonces ser un texto, número, float o cualquier cosa. En base a lo anterior, entonces podemos inferir que no es posible transformar un String a un entero, a un float o a cualquier otro tipo de datos, ya que un String puede contener, por ejemplo, “**1.23P**”, lo cual por tener una letra P ya no es un dato numérico.

Para solucionar esto se pueden hacer las transformaciones que deseemos. Por ejemplo:

“1.2” → Se puede convertir a un float o a un double, ya que 1.2 puede serlo.

“10” → Se puede convertir a un entero.

Parser o Parseo

Las transformaciones de datos que tenemos en el ejemplo anterior se llaman Parser o Parseo. El Parser es un método que tienen todas las clases Java que corresponden a tipos primitivos.

Estas clases son:

- Integer
- Double
- Float

Cada una de estas clases contiene un método que transforma un String en su tipo de dato primitivo, por ejemplo:

- Integer en un int
- Double en un double



Tipo de parser para cada clase

```
//Variables de tipo String
String stringFloat = "1.2";
String stringDouble = "2.4";
String stringInteger = "23";

//Variables numéricas
float datoFloat;
double datoDouble;
int datoInteger;

//Cada clase tiene su propio parse
datoFloat = Float.parseFloat(stringFloat);
datoDouble = Double.parseDouble(stringDouble);
datoInteger = Integer.parseInt(stringInteger);
System.out.println("Esto es un float: " + datoFloat);
System.out.println("Esto es un double: " + datoDouble);
System.out.println("Esto es un integer: " + datoInteger);
```

Ejercicio guiado



Calcular el área de un rectángulo

Para resolver el problema, una buena técnica es identificar los pasos claves que debo realizar para llegar a la solución:

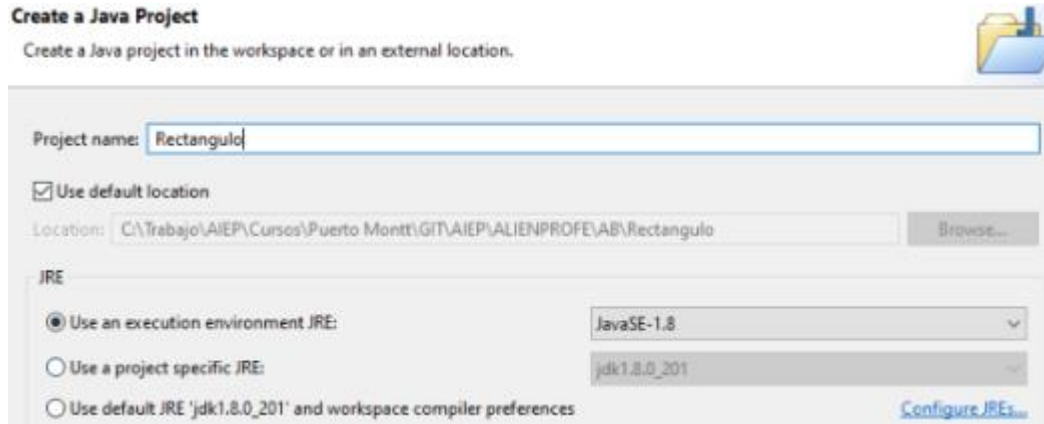
- Se necesita recibir dos datos numéricos en la entrada, correspondiente a cada lado del rectángulo.
- Cada dato debe ser mayor a cero, en el caso contrario dar al usuario un mensaje, por ejemplo, "El dato ingresado no es correcto. Ingrésele nuevamente".
- En caso de que los datos sean correctos, calcular el área con la siguiente fórmula:

$$\text{área} = L * A$$



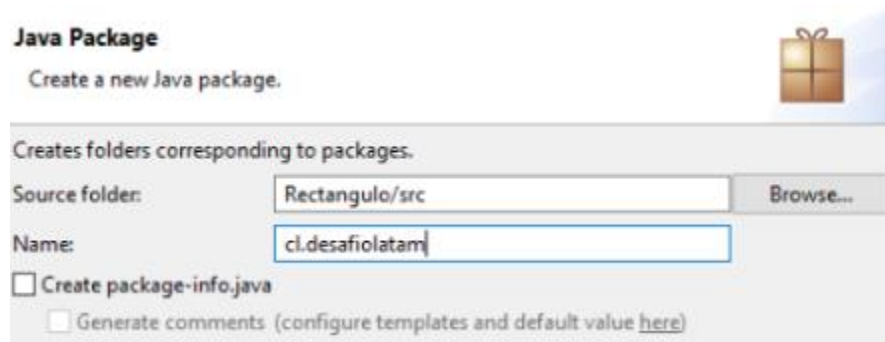
1. Crear el proyecto en Eclipse
File → New → Java Project

1. Crear el proyecto "Rectángulo"



Al crear el proyecto, se elige la estructura por defecto.

3. Crear el package **cl.desafiolatam**
src → New → Package



4. Crear la clase principal **AreaRectangulo.Java**

Clic derecho sobre el Package → New → class

Recordar:

Hacer clic en public static void main

Java Class

Create a new Java class.



Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☒ `public static void main(String[] args)`

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

5. Leer los datos: En la clase creada **AreaRectangulo** se deben leer los datos ingresados por el usuario. Considerar que los datos que se ingresen no pueden ser negativos ni tampoco igual a cero.

```
package cl.desafiolatam;
import java.util.Scanner;
public class AreaRectangulo {
    public static void main(String[] args) {
        double base = 0.0;
        double altura = 0.0;
        Scanner sc = new Scanner(System.in);
        do{
            System.out.printf("Ingrese la base: ");
            base = sc.nextDouble();
            if(base <= 0) {
                System.out.println("Dato inválido");
            }
        }while(base <= 0);
        do{
            System.out.printf("Ingrese la altura: ");
            altura = sc.nextDouble();
            if(altura <= 0) {
                System.out.println("Dato inválido");
            }
        }while(altura <= 0);
    }
}
```



6. Calcular el área

Se agregan las siguientes líneas al algoritmo:

- La primera es la declaración de la variable que guardará el resultado del cálculo.
- La segunda, será la que realizará el cálculo correspondiente.
- La tercera, la que imprimirá el resultado por pantalla.

```
double area = 0.0;  
area = base * altura;  
System.out.println("El área del rectángulo es: " + area);
```



Código completo

```
public class AreaRectangulo {
    public static void main(String[] args) {
        double base = 0.0;
        double altura = 0.0;
        double area = 0.0;
        Scanner sc = new Scanner(System.in);
        do{
            System.out.printf("Ingrese la base: ");
            base = sc.nextDouble();
            if(base <= 0) {
                System.out.println("Dato inválido");
            }
        }while(base <= 0);
        do{
            System.out.printf("Ingrese la altura: ");
            altura = sc.nextDouble();
            if(altura <= 0) {
                System.out.println("Dato inválido");
            }
        }while(altura <= 0);
        area = base * altura;
        System.out.println("El área del rectángulo es: " + area);
    }
}
```



Ejercicio propuesto



Calcular el perímetro de un rectángulo

Para resolver el problema, una buena técnica es identificar los pasos claves que debo realizar para llegar a la solución:

- Se necesita recibir dos datos numéricos en la entrada, correspondiente a cada lado del rectángulo.
- Cada dato debe ser mayor a cero, en el caso contrario dar al usuario un mensaje, por ejemplo, "El dato ingresado no es correcto. Ingrésele nuevamente".
- En caso de que los datos sean correctos, calcular el perímetro con la siguiente fórmula:

$$\text{perímetro} = 2L + 2A$$

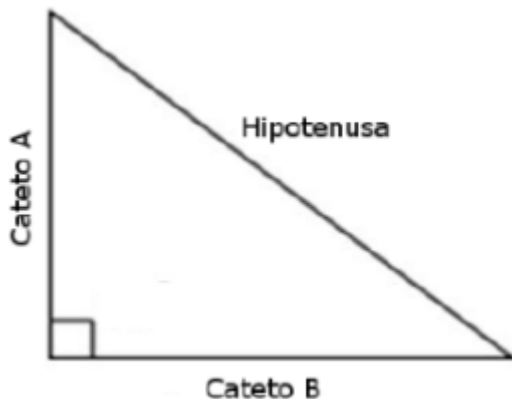


Ejercicio propuesto



Largo de la hipotenusa

La fórmula de Pitágoras nos permite calcular el largo de la hipotenusa de un triángulo rectángulo a partir de los largos de los catetos. Crearemos un programa donde el usuario introduzca los valores de ambos catetos y entreguemos como resultado el largo de la hipotenusa.



$$hipotenusa = \sqrt{a^2 + b^2}$$

¿Qué es una transformación de datos?





Próxima sesión...

Desafío guiado - Operaciones aritméticas

{desafío}
latam_

*Academia de
talentos digitales*

