Carrera de Java Programmer SE8

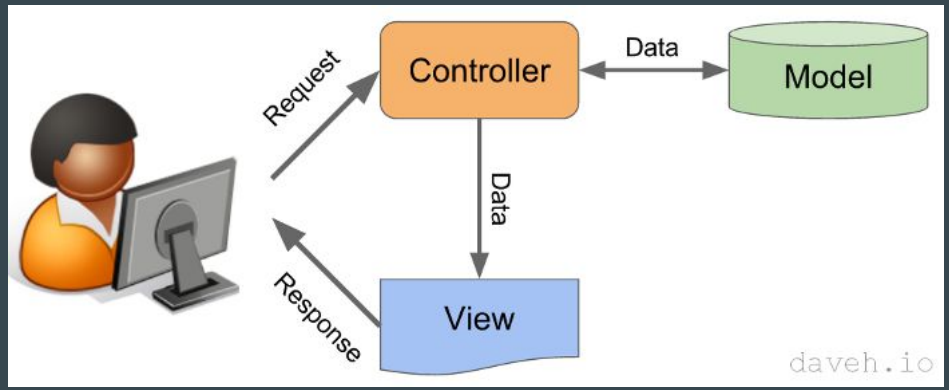
•••

Módulo 2 : Lenguaje de Programación Java

Patrón MVC

Características:

- Patrón de arquitectura de software.
- Separa los datos y la lógica de negocio.
- Se divide en tres componentes:
  - Modelo: Es la capa donde se trabaja con los datos.
  - Vista: Representa la forma de presentar los datos.
  - Controlador: Enlace entre las vistas y los modelos.



Localización

Características:

- Se encuentra en el paquete **java.util.Locale**
- Nos permite especificar la región geográfica, política y cultural.
- Soporta múltiples regiones geográficas:
  - http://www.oracle.com/technetwork/java/javase/java8locales-2095355.html
- Obtener y establecer el Locale por defecto:
  - **Locale.getDefault();**
  - **Locale.setDefault(Locale);**

Ejemplo

Internacionalización

Definición:

Capacidad de brindar a nuestra aplicación de la capacidad de estar disponible en múltiples idiomas, tomando en cuenta la Localización (Locale).

ResourceBundle:

Archivos de propiedades (recursos) de clave-valor para establecer los idiomas necesarios para nuestra aplicación.

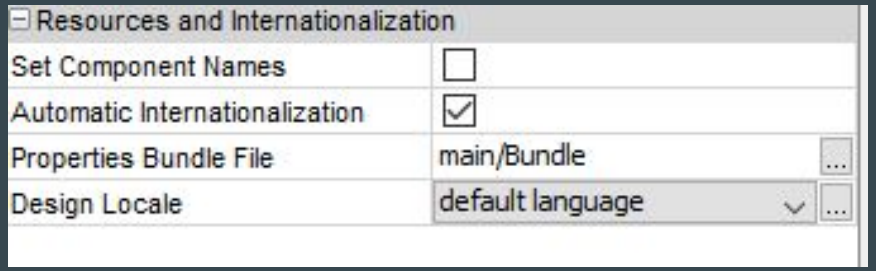
Pasos:

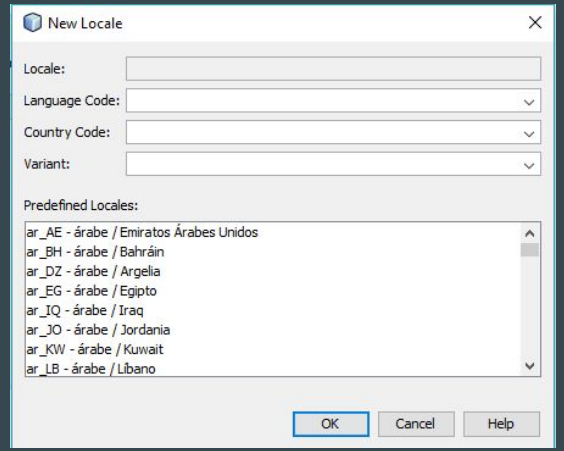
1) Crea tu proyecto con Java Swing.
2) Usa la herramienta de Netbeans para internacionalización automática.
3) Usar ***Locale*** para establecer el idioma actual.
4) Usar ***ResourceBundle*** para cambiar el idioma dinámicamente.

Internacionalización

Características:

- Automatic Internalization:
  - Propiedad boolean (true / false)
- Properties Bundle File:
  - Archivos .properties para tus idiomas
- Design Locale:
  - Idiomas disponibles para tu aplicación.





Internacionalización

Locale:

- import java.util.Locale
- Locale loc = new Locate(***IDIOMA, PAÍS***);

ResourceBundle:

ResourceBundle bundle = ResourceBundle.getBundle("RUTA_ARCHIVO");
jLabel1.setText(bundle.getString("CLAVE"));

```
Locale myLocale = Locale.getDefault();
System.out.println("My country (ISO): " + myLocale.getCountry());
System.out.println("My country name: " + myLocale.getDisplayCountry());
System.out.println("My language (code): " + myLocale.getLanguage());
System.out.println("My language (name): " + myLocale.getDisplayLanguage());
System.out.println(myLocale.getDisplayName());
```

Ejemplo

Práctica

Problema:

Crear un programa que calcule el valor de IMC (Índice de Masa Corporal) y mostrar su evaluación.

Fórmula:

- imc = p / (t * t)
- p = Peso
- t = Talla o estatura

Peso Insuficiente	< 18,5
Peso Normal	18,5 – 24,9
Sobrepeso Grado I Sobrepeso Grado II	25 – 26,9 27 – 29,9
Obesidad I Obesidad II Obesidad III (mórbida) Obesidad IV (extrema)	30 – 34,9 35 – 39,9 40 – 49,9 > 50,00

Indicaciones:

- Crear un proyecto con el nombre **IMCCalculate**
- Crear los siguientes paquetes:
  - com.main
    - EntryPoint.java
  - com.operation
    - IMCCalculate.java
  - com.structures
    - I_IMC.java

Conexión a BD con JDBC

Características:

- import java.sql.*
- Driver JDBC (Java Database Connectivity):
  - https://dev.mysql.com/downloads/connector/j/
  - Importar JAR al proyecto
  - Incluir la librería con ***Class.forName()***
- Ruta a la base de datos:
  - jdbc:<MOTOR DE BD>://<IP>:<PUERTO>/<NOMBRE BD>

Conexión a BD con JDBC

Clases:

- Connection:
  - getConnection(RUTA, USUARIO, CLAVE)
- Statement:
  - executeQuery(SQL) -> Operaciones de consulta de datos
  - executeUpdate(SQL) -> Operaciones para ingresar, actualizar o eliminar datos
  - execute(SQL) -> Múltiples resultados
    - true -> ResultSet object;
    - false -> Sin resultados
- ResultSet:
  - Objeto complejo para el manejo de datos.

Ejemplo