	SNIR 1	
	Programmation objet :	IHM
	~ IHM en JAVA	

# TP IHM JAVA

---

## 1 OBJECTIF

---

- Mettre en œuvre quelques composants et containers d'interface graphique
- Prendre en compte les événements générés par les composants

## 2 MISE EN SITUATION

---

A votre disposition le fichier `dietetique.jar` contenant la classe de votre application.

## 3 RÉALISATION

---

Vous pouvez avoir une idée de l'attendu de votre travail à l'aide de l'archive JAVA **dietetique.jar** fournie en ressource du TP

Pour l'exécuter, en ligne de commande, vous vous placez dans le répertoire contenant l'archive et vous lancez votre application Java comme suit :

**java -jar dietetique.jar**

Pour obtenir cette application, vous utiliserez Netbeans.

- Créez un nouveau projet noté **dietetique** puis un package **diet**
  - Dans le package **diet**, créez une **classe visuelle** nommée **PoidsIdeal** de type **Jframe**
-

### 3.1 Fenêtre principale

#### 3.1.1 Caractéristiques principales

Cette fenêtre aura à minima les caractéristiques suivantes

- taille 530 × 620
- non redimensionnable

#### 3.1.2 composant de la fenêtre

- Vous positionnerez le gestionnaire de placement de la JFrame à *null*.

**1. Sur le panneau, vous ajouterez les composants suivants (respectez les noms) :**

```
-JButton jButtonDevine
-JButton jButtonLorentz
-JButton jButtonLorentzAge
-JButton jButtonSuite
-JLabel jLabelAge
-JLabel jLabelNom
-JLabel jLabelPoids
-JLabel jLabelPrenom
-JLabel jLabelSexe
-JLabel jLabelTaille
-JPanel jPanelInformations
-JPanel jPanelPoidsIdeal
-JPanel jPanelSexe
-JRadioButton jRadioFemme
-JRadioButton jRadioHomme
-JSpinner jSpinnerAge
-JSpinner jSpinnerPoids
-JSpinner jSpinnerTaille
-JTextArea jTextAreaAfficheur
-JTextField jTextFieldNom
-JTextField jTextFieldPrenom
```

The diagram illustrates the layout of the main window. It features a form with the following elements:

- Nom** and **Prénom**: Text input fields.
- Sexe**: Radio buttons for **Femme** and **Homme**.
- Age**: A spinner field with a value of 1.
- Poids**: A spinner field with a value of 15.
- Taille**: A spinner field with a value of 0,5.
- Suite**: A button below the form fields.
- Poids idéal**: A section containing three buttons: **Poids idéal devine**, **Poids idéal lorentz**, and **Poids idéal lorentz / age**.
- jPanelSexe**: A callout pointing to the radio buttons.
- jPanelInformations**: A callout pointing to the form fields.
- jPanelPoidsIdeal**: A callout pointing to the 'Poids idéal' section.

Vous mettrez les gestionnaires de placement les mieux adaptés pour chaque élément de type JPanel.

Les éléments de type JSpinner peuvent être modélisés pour n'autoriser qu'un type particulier de données.

Ainsi, JSpinnerAge ne doit prendre en compte que des données de type entier comprises entre 1 et 200 et le pas de progression sera de 1.

JSpinnerPoids ne doit prendre en compte que des données de type double comprises entre 15 et 300 et le pas de progression sera de 0,1.

jSpinnerTaille ne doit prendre en compte que des données de type double comprises entre 0,5 et 3,0 et le pas de progression sera de 0,01.

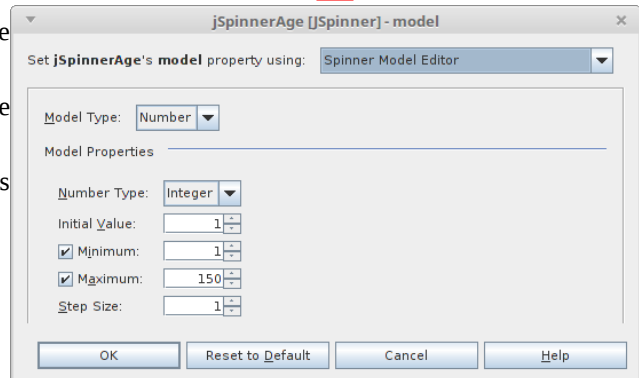
Pour modifier ses propriétés, il faut aller dans le panneau propriétés de l'élément et cliquez sur le bouton à droite de la propriété model :



Choisissez un modèle de type "Number", puis Integer ou Double.

Positionnez la valeur initiale, le minimum, le maximum et la valeur du pas.

**Attention** pour les réels, les valeurs sont de la forme xxx,xxx et non xxx.xxx .



## 3.2 Gestion des événements

### 2. Ajoutez les données membres suivantes à la classe PoidsIdeal.

```
private String nom;
private String prenom;
private double imc;
private double poids;
private double taille;
private int age;
final private double imcs[] = {16.5, 18.5, 25, 30, 35, 40};
final private String corpulences[] = {"Famine", "Maigreur", "Normale", "Surpoids", "Obésité modérée", "Obésité sévère", "Obésité morbide"};
final private int NBIMC = 6;
final private int NBCORPULENCE = 7;
```

### 3.2.1 Le bouton « Suite »

Lorsque l'utilisateur clique sur le bouton « suite », les actions suivantes doivent être effectuées :

- Le nom et le prénom se trouvant respectivement dans jTextFieldNom et jTextFieldPrenom initialiseront les attributs nom et prenom.
- La valeur de jSpinnerAge initialisera l'attribut age.
- Les valeurs de jSpinnerPoids et jSpinnerTaille initialiseront les attributs poids et taille
- L'attribut imc sera initialisé par le calcul de ce dernier
- Un message de bienvenue rappelant le nom et le prénom de l'utilisateur sera affiché dans l'élément jTextAreaAfficheur.
- La valeur de son imc et la corpulence associée seront affichées dans l'élément jTextAreaAfficheur.

Voici une partie du code correspondant :

```
poids = (double) jSpinnerPoids.getValue();
age = (int) jSpinnerAge.getValue();
nom = jTextFieldNom.getText();

jTextAreaAfficheur.append("\nBonjour " + prenom + " " + nom);
jTextAreaAfficheur.append("\nVotre indice de masse corporel est de " + imc);
// recherche de la corpulence de l'individu en fonction de son imc
int indiceCorpulence = 0;
for (int i = 0; i < NBIMC - 1; i++) {
    if (imc > imcs[i] && imc <= imcs[i + 1]) {
        indiceCorpulence = i + 1;
    }
}
// cas extreme
if (imc < 16.5) {
    indiceCorpulence = 0;
}
if (imc > 40) {
    indiceCorpulence = NBCORPULENCE - 1;
}
jTextAreaAfficheur.append("\nVotre corpulence est qualifiée de " + corpulences[indiceCorpulence]);
```

**3. Associez l'événement clique du bouton « Suite » avec le code correspondant (le code fourni est à compléter selon les actions décrites plus haut).**

Rappel :

$$imc = \frac{poids}{taille^2}$$

### 3.2.2 Les boutons de calcul de poids idéal

On fourni le code correspondant au calcul du poids idéal selon la formule de Devine :

```
double poidsIdeal = 0;
if (jRadioFemme.isSelected()) {
    poidsIdeal = 45.5 + 2.3 * (taille / 0.0254 - 60.0);
}
if (jRadioHomme.isSelected()) {
    poidsIdeal = 50.0 + 2.3 * (taille / 0.0254 - 60.0);
}
if (poidsIdeal > 0) {
    jTextAreaAfficheur.append("\nVotre poids ideal avec la formule
de devine : " + ((Double) poidsIdeal).toString() + " kg\n");
    double ecart = poids - poidsIdeal;
    if (ecart >= 0) {
        jTextAreaAfficheur.append("\nVous devez perdre " + ((Double)
ecart).toString() + " kg\n");
    }
    else {
        jTextAreaAfficheur.append("\nVous devez prendre " +
((Double) (-ecart)).toString() + " kg\n");
    }
}
```

**4. Associez ce code à l'événement correspondant pour le bouton jButtonDevine.**

**5. Faites de même pour les deux autres calculs de formule possible.**

Rappel des formules :

Formule du poids idéal selon Lorentz.

$$\text{Pour une femme : poids ideal} = \text{taille} \times 100 - 100 - \frac{(\text{taille} \times 100 - 150)}{2.5}$$

$$\text{Pour un homme : poids ideal} = \text{taille} \times 100 - 100 - \frac{(\text{taille} \times 100 - 150)}{4}$$

Formule du poids idéal selon Lorentz en tenant compte de l'âge, mais pas du sexe.

$$\text{Poids ideal} = 50 + \frac{(\text{taille} \times 100 - 150)}{4} + \frac{(\text{age} - 20)}{4}$$

Formule du poids idéal selon Devine.

$$\text{Pour une femme : poids ideal} = 45,5 + 2,3 \times \left( \frac{\text{taille}}{0,0254} - 60 \right)$$

$$\text{Pour un homme : poids ideal} = 50 + 2,3 \times \left( \frac{\text{taille}}{0,0254} - 60 \right)$$

**6. Complétez votre code afin que seul le panneau d'informations soit visible au lancement de l'application.**

**7. Complétez votre code afin d'avoir le comportement suivant lors d'un clic sur le bouton "suite" :**

**Si le panneau d'information était visible, ce dernier disparaît et le panneau de calcul du poids idéal apparaît.**

**Inversement, si le panneau d'information était invisible, ce dernier apparaît et le panneau de calcul du poids idéal disparaît.**