Module 1 –Informatique – Algorithmique et Programmation Objet
Travaux Pratiques (14), Licence 1ère Année
Swing – Les Gestionnaires de Placement

Exercice 1 Positionnement de composants sans Layout

Il est possible, bien que fastidieux et non recommandé, de positionner soi-même les composants dans un conteneur (**JFrame**, **JPanel**, …).

Reprenez l'exemple du **TP 13** sur l'introduction de Swing.

Dans le constructeur de la classe **MaFenetre**, apportez les modifications suivantes :

- Supprimez la stratégie de placement **FlowLayout** appliquée au **ContentPane** de la fenêtre (**contentPane.setLayout(null) ;)**
- Décomposez la création, l'initialisation et l'ajout de chaque composant dans le **ContentPane** ainsi :

  contentPane.add(new JButton("Poussez-moi")) ;

  devient

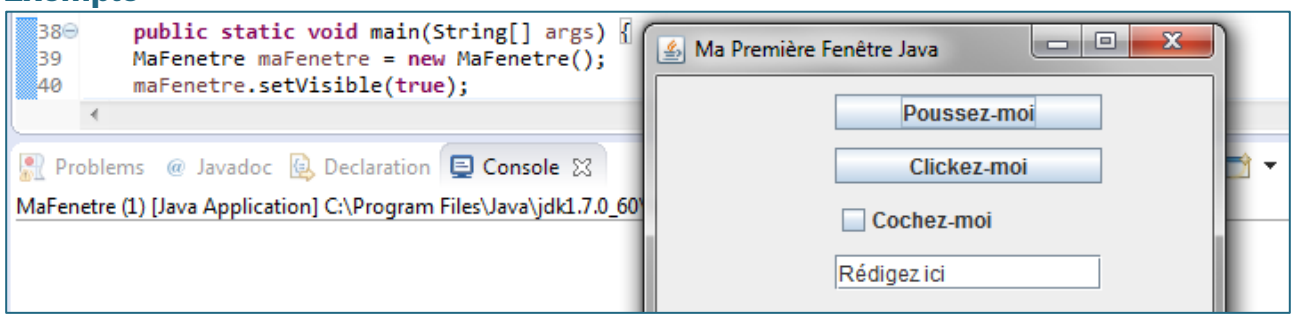
  JButton btPousseMoi = new JButton("Poussez-moi") ;
  btPousseMoi.setBounds(100, 30, 150, 50) ;
  contentPane.add(btPousseMoi) ;

  Note : **setBounds()** permet de définir la position et la taille du composant. Dans l'exemple ci-dessus, **btPousseMoi** sera décalé de **100** pixels depuis la gauche, de **10** pixels depuis le haut, aura une longueur de **150** pixels sur **20** pixels de hauteur.

  Répétez la même opération pour les trois autres composants (**JButton**, **JCheckBox** et **JTextField**).

Exécutez votre **main()**.

Exemple



Remarque : si vous réduisez la taille de la fenêtre, les composants ne sont pas repositionnés et peuvent ne plus être visibles. Testez.

Vous pouvez tester l'ajout d'autres composants à des positions différentes.

Exercice 2 Positionnement de composants avec le GridLayout

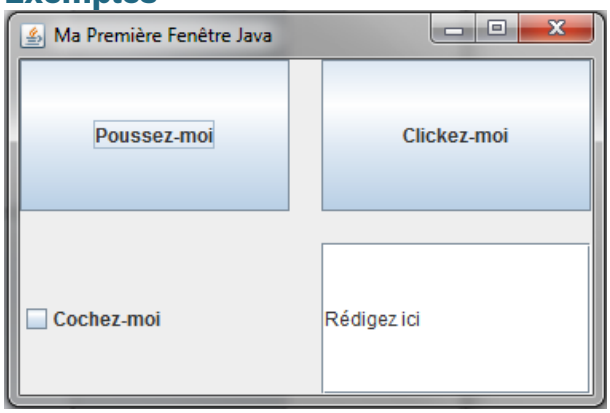
Avec le **GridLayout**, il faut définir le nombre de lignes et de colonnes (cellules) que l'on souhaite pour le conteneur. Les composants sont placés par défaut dans les cellules de gauche à droite et du haut vers le bas.

Note : dans ce cas, les indications de tailles (**setSize()** et **setBounds()**) ne sont pas prises en compte.
On peut cependant définir l'espacement, verticalement et horizontalement, entre les cellules.

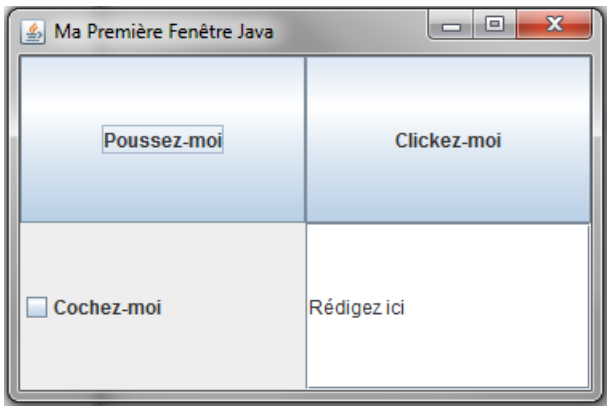
- indiquez l'application du gestionnaire de placement **GridLayout** au **ContentPane** de la fenêtre (**contentPane.setLayout(new GridLayout(2, 2, 20, 20)) ;**)
  Note : les deux valeurs **2** indiquent respectivement le nombre de ligne et le nombre de colonnes. Les deux valeurs **20** indiquent respectivement l'espace horizontal et vertical entre les cellules. Ces deux derniers peuvent être omis si l'on ne souhaite pas d'espacement entre les cellules.
  Remarque : l'appel de la méthode **setBounds()** est ineffectif et peut être supprimé. Le composant occupe toute la cellule où il est placé.

Exécutez votre **main()**.

Exemples



Avec espacement entre cellules

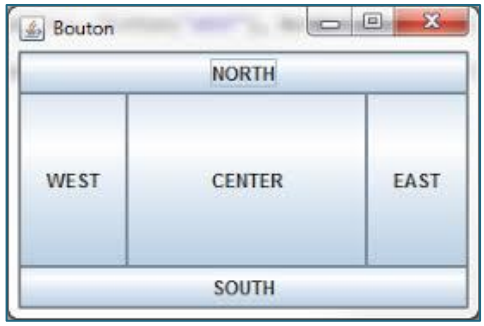


Sans espacement entre cellules

Testez d'autres configurations (4 lignes x 1 colonne, 1 ligne x 4 colonnes, …).

Exercice 3 Positionnement de composants avec le BorderLayout

Le gestionnaire **BorderLayout** définit 5 zones de placement tel que décrit dans la figure ci-dessous : une zone au Nord (**NORTH**), une au Sud (**SOUTH**), une à l'Est (**EAST**), une à l'Ouest (**WEST**) et la dernière au centre (**CENTER**).



Zones de placement du gestionnaire BorderLayout

- indiquez l'application du gestionnaire de placement **BorderLayout** au **ContentPane** de la fenêtre (**contentPane.setLayout(new BorderLayout() ;**)

Le gestionnaire **BorderLayout** requiert une contrainte lors de l'ajout d'un composant. La contrainte doit être une des suivantes :

- **BorderLayout.NORTH**
- **BorderLayout.SOUTH**
- **BorderLayout.EAST**
- **BorderLayout.WEST**
- **BorderLayout.CENTER**

Ces contraintes sont passées en 2ème paramètre de la méthode **add()** qui permet de rajouter un composant à un conteneur.

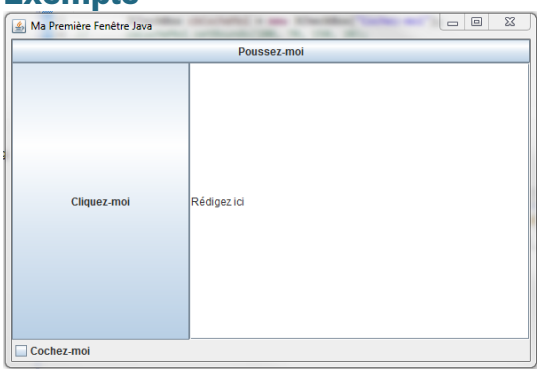
- Modifiez votre constructeur de la class **MaFenetre** de façon à placer vos composants comme suit :
  - Le composant **btPousseMoi** dans la zone Nord du **BorderLayout**
  - Le composant **btCliqueMoi** dans la zone Ouest du **BorderLayout**
  - Le composant **btCocheMoi** dans la zone Sud du **BorderLayout**
  - Le composant **tfRedigeMoi** dans la zone centre du **BorderLayout**

Remarque : certaines contraintes de taille peuvent être prises en compte par le **BorderLayout**.

- Indiquez une largeur préférée pour le composant **btCliqueMoi** de façon qu'elle soit de **200** pixels de largeur (**btCliqueMoi.setPreferredSize(new Dimension(200, 0)) ;**)
  Note : la méthode **setPreferredSize** nécessite un paramètre objet **Dimension** qui lui-même accepte 2 paramètres (la largeur et la hauteur).
  Remarque : vous noterez que dans ce cas, seule la largeur est prise en compte. La hauteur est déterminée par les hauteurs des 2 composants au Nord et au Sud.

Exécutez votre **main()**.

Exemple



Remarque : dans notre exemple, la zone Est n'est pas utilisée, elle est donc réduite à 0 x 0 pixel.

Note : On peut ajouter un conteneur dans un conteneur.

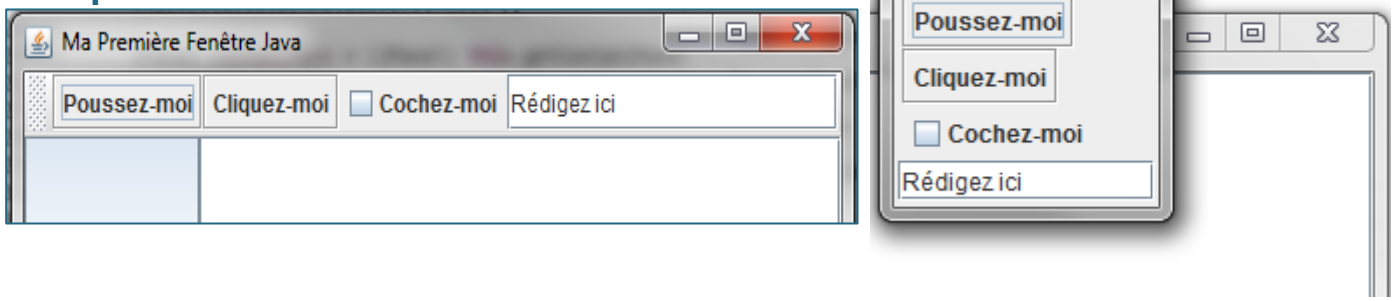
Nous souhaitons que :

- La zone Nord du **BorderLayout** représente une barre d'outils :
  Note : **JToolBar** est un conteneur qui peut être ajouté à un autre conteneur.

- Dans votre Classe **MaFenetre** écrire une méthode privée **JToolBar creerBarreOutils()** qui crée un conteneur **JToolBar** qui contiendra les mêmes 4 composants déjà définis (**PousseMoi**, **CliqueMoi**, **CocheMoi** et **RedigeMoi**) et le retourne comme résultat.
- Placez la **JToolBar** créée dans la zone Nord du **BorderLayout** associé à votre fenêtre :
  (**contentPane.add(creerBarreOutils(), BorderLayout.NORTH) ;**)

Exécutez votre **main()**.

Exemple



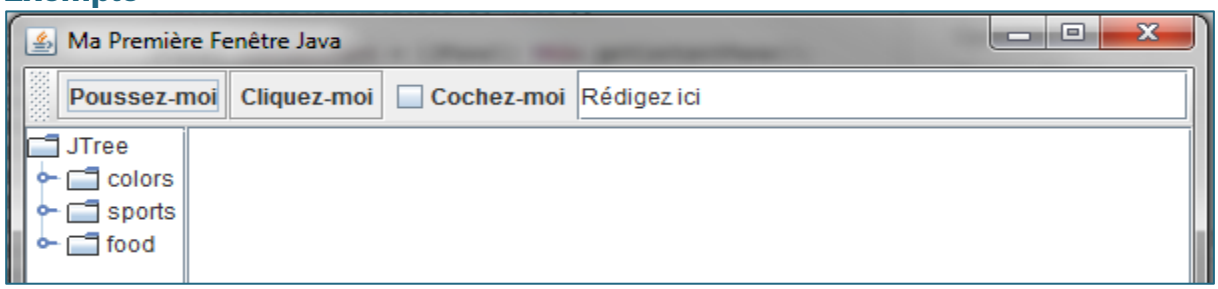
Barre déplacée

Nous souhaitons maintenant que :

- La zone Ouest du **BorderLayout** représente une arborescence :
  Note : **JTree** est un composant qui permet de présenter les données sous forme hiérarchique arborescente (arbre).
  JScrollPane est un conteneur permettant de munir un composant de barres de défilement, comme un composant **JTree**.
  - Dans votre constructeur de la Classe **MaFenetre**, supprimez le composant **btCliqueMoi** et créez un conteneur **JScrollPane** auquel vous passerez un composant **JTree**.
    (**JScrollPane composantOuest = new JScrollPane(new JTree()) ;**)
  - Placez l'instance **composantOuest** dans la zone Ouest du **BorderLayout** associé à votre fenêtre :
    (**contentPane.add(composantOuest, BorderLayout.WEST) ;**)

Exécutez votre **main()**.

Exemple



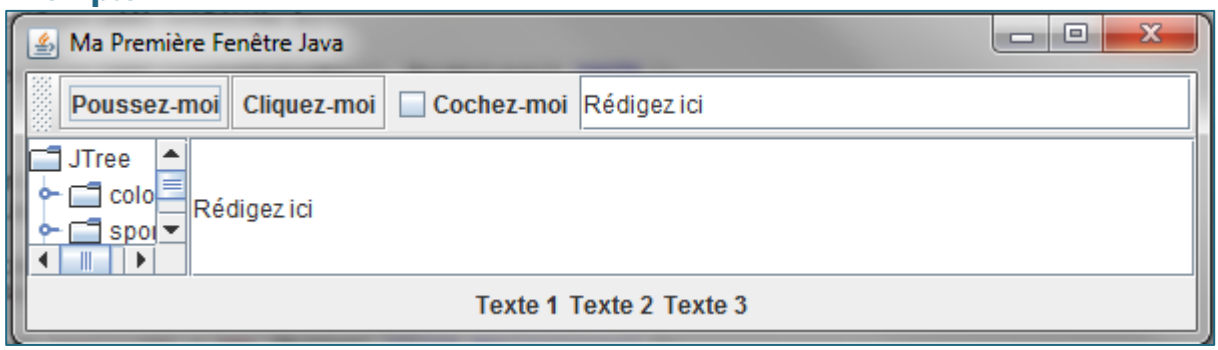
Remarque : vous pouvez définir une largeur préférée (**setPreferredSize**) pour le composant **composantOuest** si la largeur actuelle n'est pas suffisante.
Nous souhaitons maintenant que :

- La zone Sud du **BorderLayout** représente une barre d'états :
  Note : On peut utiliser un objet **JPanel**, qui contiendra plusieurs composants, pour définir la barre d'état.

- Dans votre Classe **MaFenetre** écrire une méthode privée **JPanel creerBarreEtats()** qui crée un conteneur **JPanel** qui contiendra 3 composants **JLabel** et le retourne comme résultat. La stratégie **FlowLayout** sera appliquée au conteneur **JPanel** retourné.
  Note : un **JLabel** est un composant permettant d'afficher du texte.
- Dans votre constructeur de la Classe **MaFenetre**, supprimez le composant **btCocheMoi** et remplacez-le par la barre d'états créée.
  (**JPanel barreEtats = creerBarreEtats()) ;**)
- Placez l'instance **composantSud** dans la zone Sud du **BorderLayout** associé à votre fenêtre :
  (**contentPane.add(barreEtats, BorderLayout.SOUTH) ;**)

Exécutez votre **main()**.

Exemple



Remarque : vous pouvez définir un placement gauche ou droit pour les composants de la barre d'états.

Nous allons ajouter un menu à la fenêtre mais celle-ci a son **contentPane** rempli. Nous allons donc le placer dans le **menuBar** de notre fenêtre. Pour cela nous utiliserons les classes :

JMenuBar : encapsule une barre de menus,
JMenu : encapsule un menu,
JMenuItem : encapsule un élément d'un menu.

- Créer un objet **JMenuBar** (**JMenuBar menuBar = new JMenuBar();**) et ajoutez-le au **menuBar** de la fenêtre : (**setJMenuBar(menuBar);**).
- Créer un objet **JMenu** avec le paramètre "**Fichier**" (**JMenu menu1 = new JMenu("Fichier");**),
- Créer deux objets **JMenuItem** avec chacun le paramètre "**Ouvrir**" et "**Fermer**" (**JMenuItem item11 = new JMenuItem("Ouvrir");**) ,
- Placer les deux objets **JMenuItem** dans l'objet **JMenu** précédent (**menu1.add(item11);**) et placer l'objet **JMenu** dans l'objet **JMenuBar** (**menuBar.add(menu1);**).

Procéder de la même façon pour ajouter un 2ème menu avec respectivement les paramètres "Edition", "**Copier**" et "**Coller**".

Exemple