

INTRODUCTION AU JAVA GRAPHIQUE AVEC NETBEANS

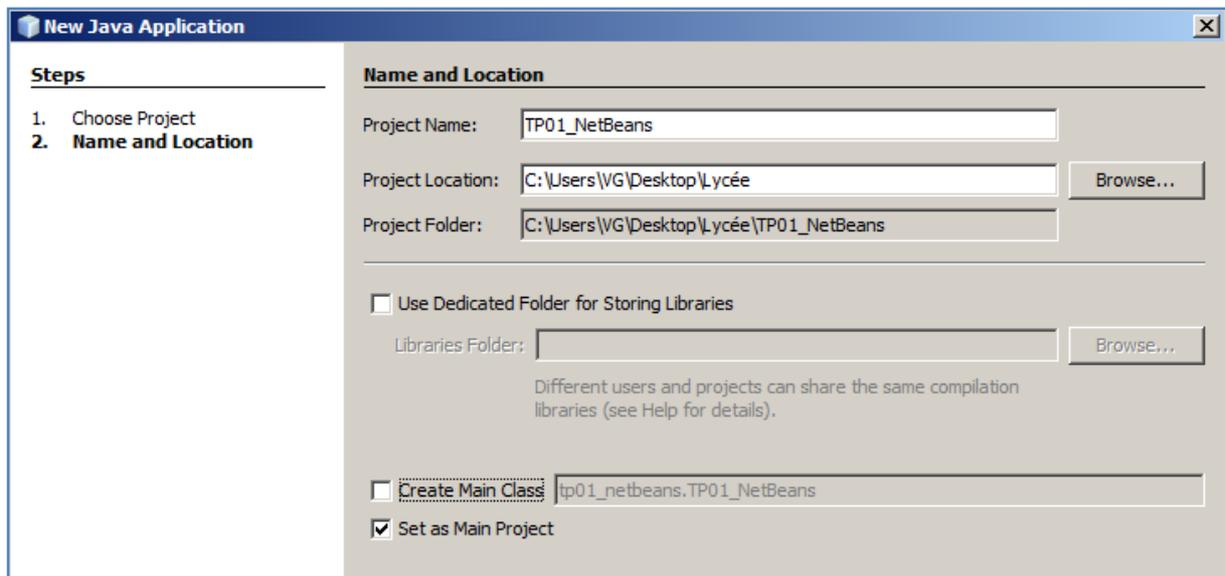
LES FICHIERS EXECUTABLES (.JAR) DES EXERCICES SONT FOURNIS. POUR LES LANCER POUR UN DOUBLE-CLIC, CHOISISSEZ DANS "OUVRIR AVEC" JAVAW.EXE DU JRE/BIN.

1. ADDITION DE 2 NOMBRES

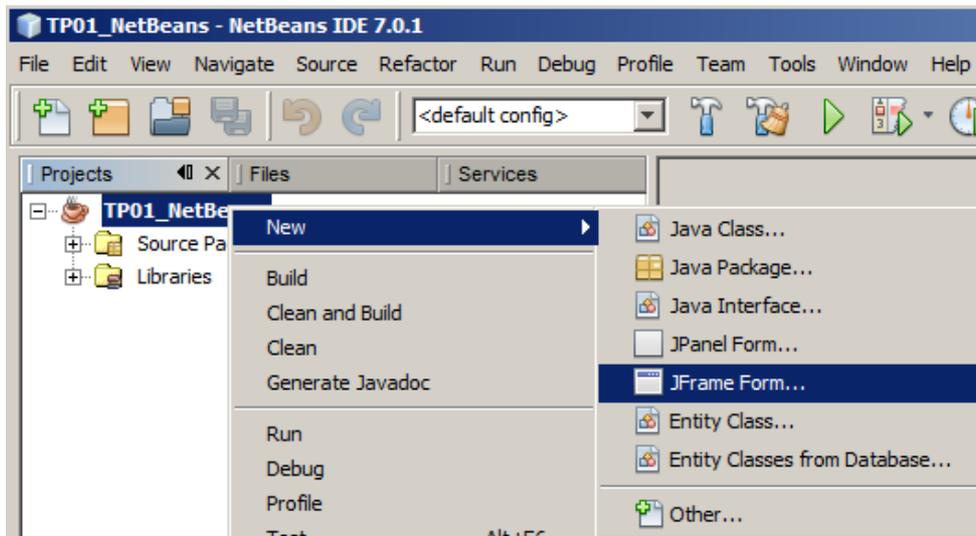
Partie 1 – Création de l'interface

1. Ouvrez NetBeans et créer un nouveau projet nommé **TP01_NetBeans** (File/New Projet/JavaApplication).

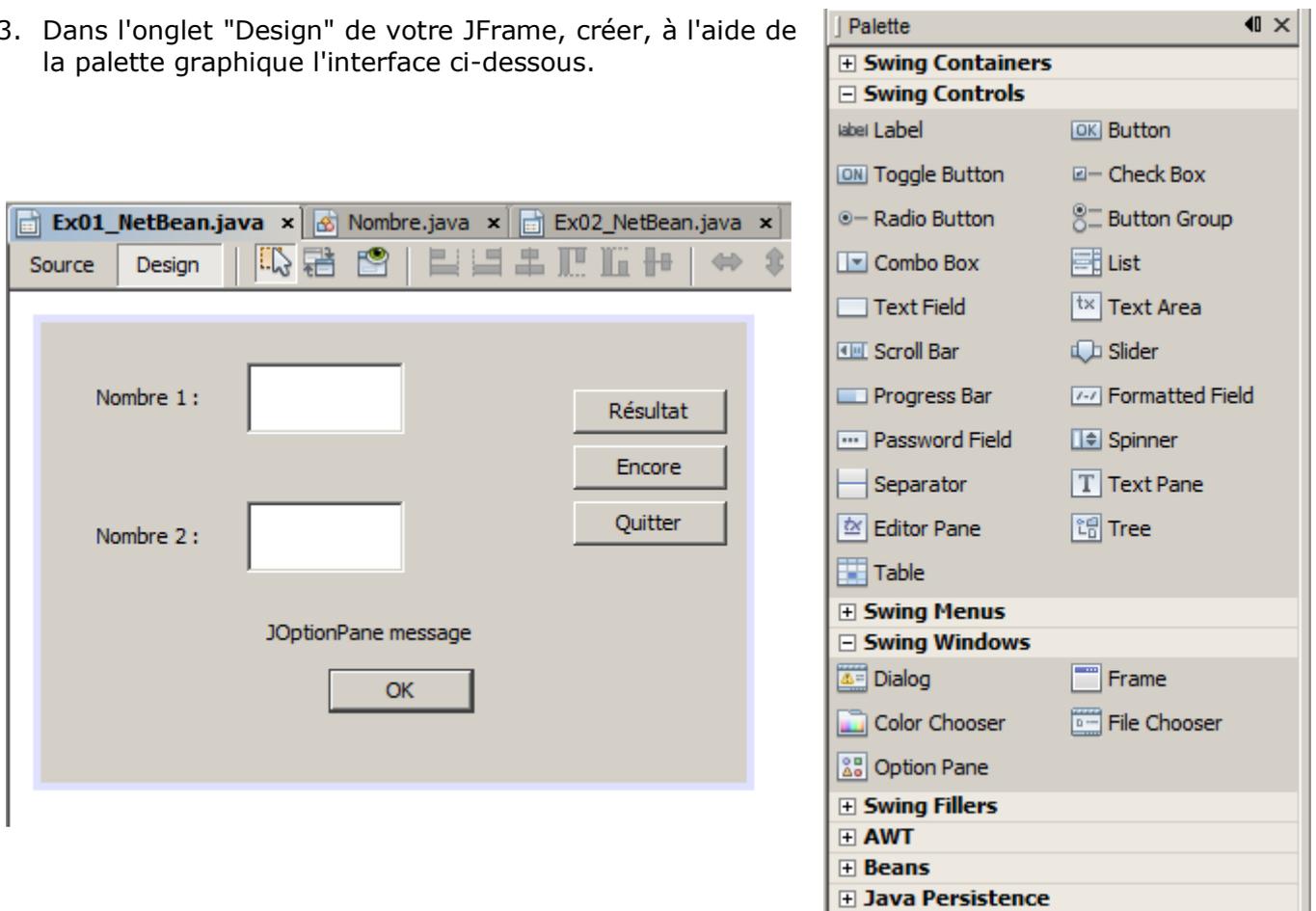
Prenez soin de **décocher** "Create Main Class" et de **cocher** "Set as Main Project".



2. Dans ce projet créez, dans une nouvelle fenêtre nommée **Ex01_NetBeans** (New/JFrame Form).



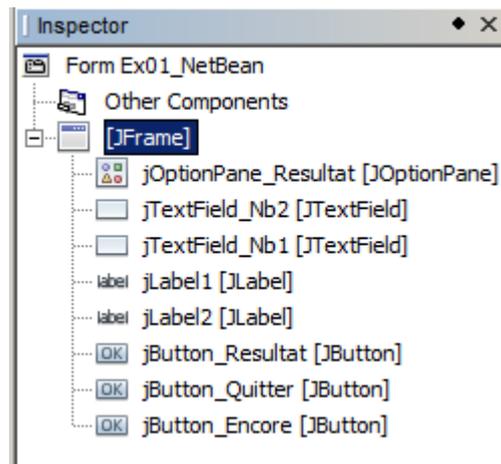
3. Dans l'onglet "Design" de votre JFrame, créer, à l'aide de la palette graphique l'interface ci-dessous.



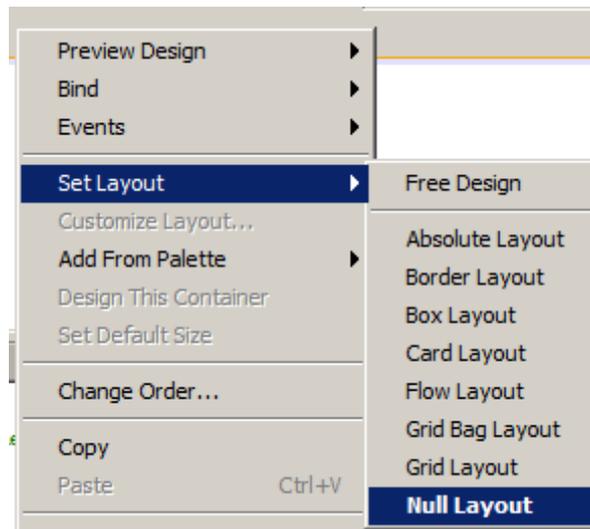
Afin de pouvoir programmer correctement et de retrouver facilement vos différentes contrôles graphiques , vous veillerez à les renommer selon la convention de nommage suivante :

TypeContrôle_NomContôle

Ex: JOptionPane1 devient JOptionPane_Resultat, JTextField_2 devient JTextField_Nb2
 Vous devez obtenir ceci dans la vue Inspector



NB : Vous pouvez si vous le voulez empêcher le placement automatique de vos objets graphiques. Pour cela, cliquer droit sur votre JFrame, **SetLayout/Null Layout**.



Partie 2 – Etude du code généré.

4. Dans l'onglet "Source" de votre JFrame, observez le squelette de code généré automatiquement par NetBeans et répondez aux questions suivantes :

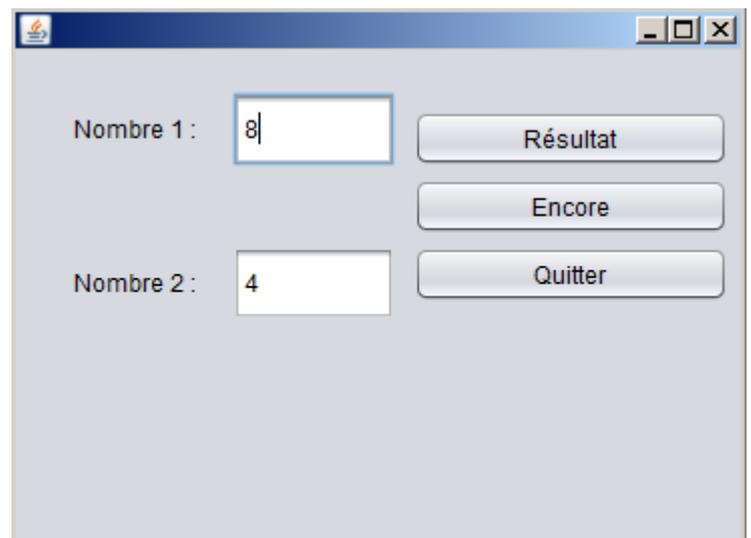
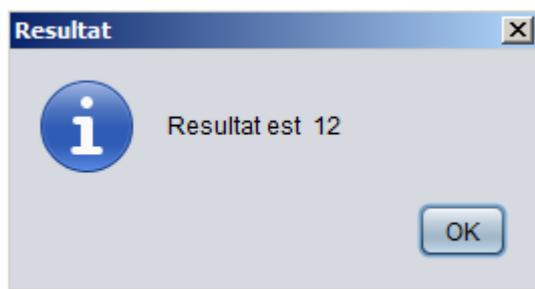
- **De quelle classe hérite votre classe JFrame nommée Ex1_NetBeans ?**
- **Votre classe JFrame possède-t-elle un constructeur ?**
- **La méthode initComponents() est-elle une procédure ou une fonction ?**
- **Si vous observez le code, vous allez découvrir un rectangle contenant les mots "Generated Code". Déroulez le code en cliquant sur le signe "+" et observez-le. Que cache cette partie ?**

```
@SuppressWarnings("unchecked")
Generated Code
```

- **Peut-on dire que le code généré par NetBeans est lié à de la programmation événementielle ? Justifiez votre réponse.**
- **Voici le code du bouton Quitter. Que représente le paramètre evt ?**
- **Votre classe JFrame possède-t-elle un programme principal ?**
- **A quoi sert la méthode *public void run()* ?**
- **Vos contrôles graphiques constituent-ils des propriétés privées ou des variables publiques.**

Partie 3 – Codage partiel de l'application.

5. Dans l'onglet "Source" codez le corps de la méthode correspondant aux boutons **Quitter** (qui ferme l'application) et **Résultat** (qui affiche le résultat de l'addition des 2 nombres saisis par l'utilisateur dans une boîte de dialogue). Testez votre application.



Partie 4 – Création d'une classe Nombre.

6. Dans votre projet, créez un nouveau fichier Java nommé Nombre.java.
7. Ecrivez le code de la classe Nombre qui offre comme service la génération d'un nombre aléatoire compris **entre 1 et 10**.

Nombre	
-	int nb
+	Nombre()
+	int get_Nombre()

***NB** : Vous utiliserez la classe Math proposant les méthodes round (qui arrondit un réel à l'entier le plus proche) et random (qui génère un réel compris entre 0 et 1)*

8. Modifier le code de votre classe JFrame afin que l'utilisateur n'ait plus à saisir de nombre dans les zones de texte à l'ouverture de la fenêtre. Vous ferez appel à la classe Nombre pour effectuer cette modification.

Partie 5 – Codage final de l'application.

9. Codez le corps de la méthode correspondant au bouton **Encore**. Il modifie les nombres des 2 zones de texte par appel de la classe Nombre. Testez votre application.

Partie 6 – Modification de l'application.

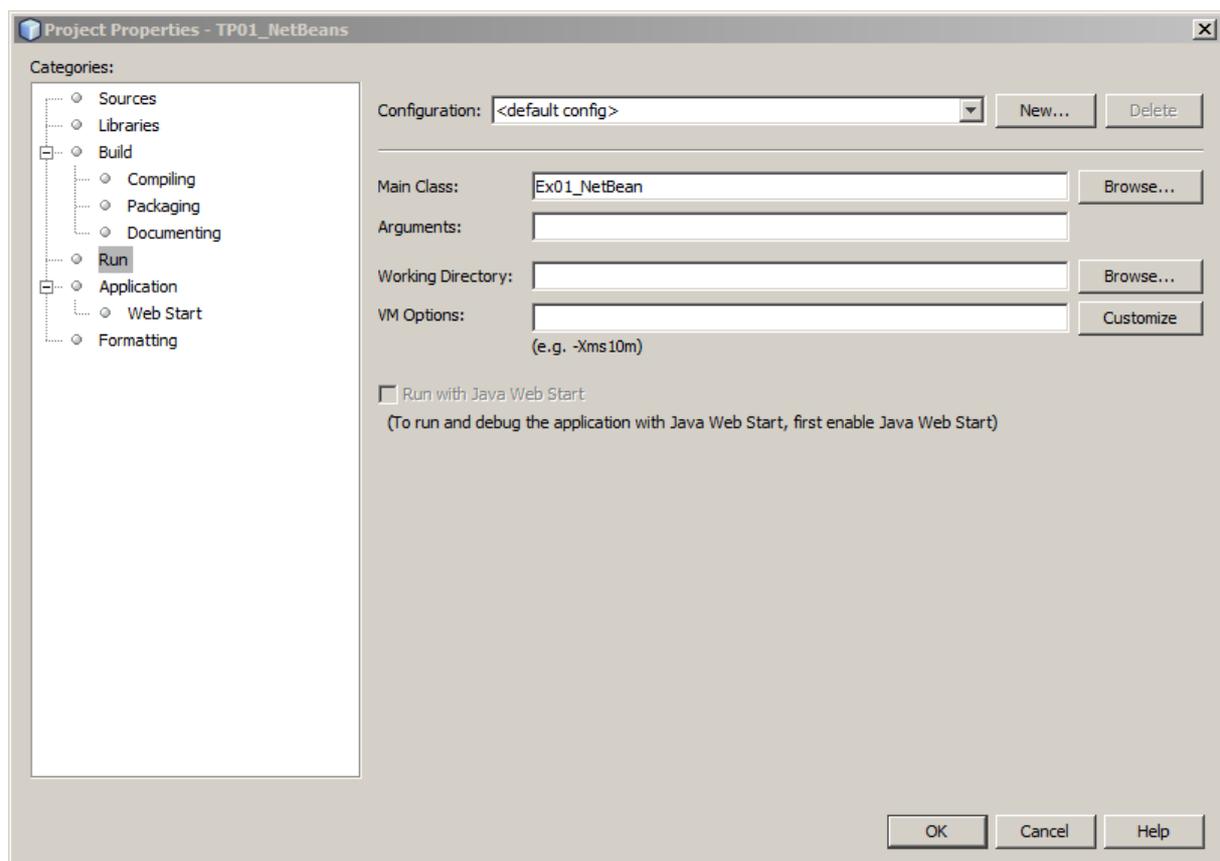
10. Dupliquez votre 1^{er} exercice **Ex01_NetBeans.java** à l'aide de l'option Refactor/Copy dans le fichier **Ex02_NetBeans.java**.

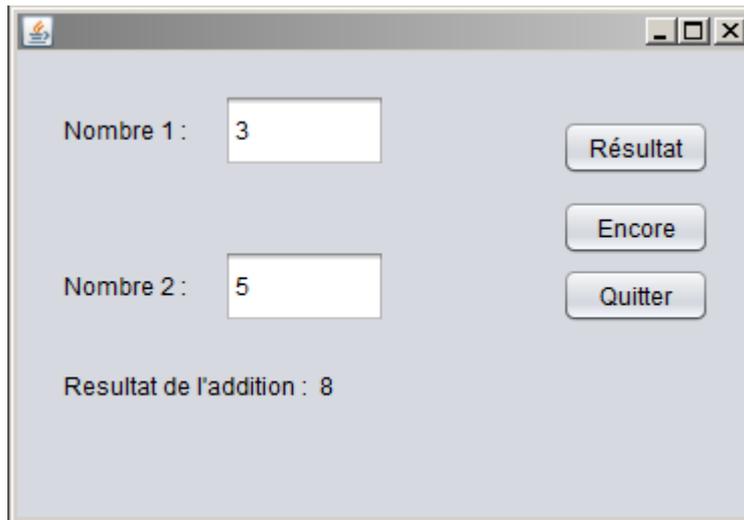
***NB** : Le refactoring est une option offrant différentes fonctionnalités comme Renommer ou déplacer des classes, méthodes et/ou attributs. Le refactoring est une option puissante qui va analyser le code source et renommer les signatures de méthodes afin qu'elle soit en correspondance avec le nom du fichier Java.*

11. Dans l'onglet "Design" de votre fichier **Ex02_NetBeans.java**, supprimez votre boîte de dialogue et remplacez-la par un label nommé **jLabel_Résultat**. Modifier votre code de manière à ce que le résultat de l'addition s'y affiche.

12. Testez votre application en prenant de soin de :

- Vérifier dans votre programme principal que la méthode run() rend bien visible votre JFrame.
- Modifier dans les propriétés de votre projet (rubrique Run/MainClass) que le fichier à exécuter est bien **Ex02_NetBeans**.





***NB** : Vous pouvez aussi lancer l'exécution d'un de vos exercices (sans passer par la fenêtre de propriété) en utilisant l'option **Run File** accessible par le clic droit sur le fichier JAVA à exécuter.*

Partie 7 – Lancement du fichier exécutable de votre application.

*NetBeans transforme vos fichiers JAVA en fichier exécutable CLASS. Il génère également un fichier exécutable de votre projet nommé **TP01_NetBeans.JAR** situé dans le répertoire "dist". C'est le fichier précisé dans la rubrique "Run" des propriétés du projet qui sera celui exécuté par l'exécutable (au cas où vous avez dans un projet plusieurs fichiers contenant un main class() comme c'est le cas ici).*

13. Recherchez sur votre machine le fichier JAVA.EXE qui a pour rôle d'exécuter des fichiers JAR. Copiez-le sur le répertoire "dist" de votre projet. Dans une fenêtre de commande, placez-vous dans le répertoire "dist" de votre projet puis lancez l'exécution de **Ex1_NetBeans** à l'aide de la commande ci-dessous :

Invite de commande> java -jar exécutable.jar

```
C:\Users\UG\Desktop\Lycée\S3\NetBeans\TP01_NetBeans\dist>java -jar TP01_NetBeans.jar
```

14. Faites de même avec **Ex2_NetBeans**.