

**NFP121 : Programmation avancée  
TP n° 1 - L'environnement de  
programmation BlueJ**

prerequis

**Installation du JDK et de BlueJ**

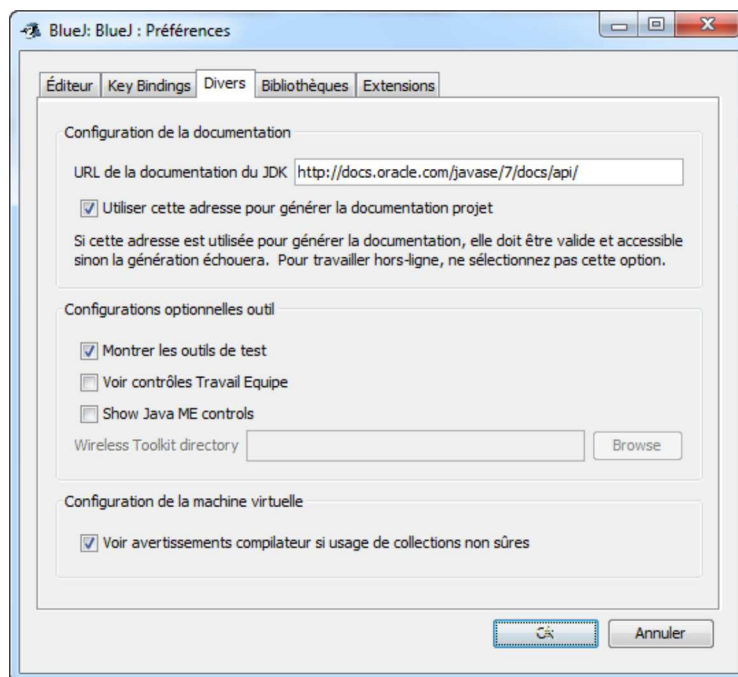
- Installation du [JDK 7](#)
- Installation de [BlueJ](#)
- Un navigateur avec une version 1.7 ou + de la JVM

**remarques :**

Pour obtenir 'Bluej' en français

- éditer le fichier '**bluej.defs**' du répertoire **../blues/lib**
- remplacer dans ce fichier ce fichier : **bluej.language=english** par **#bluej.language=english**
- et **#bluej.language=french** par **bluej.language=french**

Pour activer les outils de tests unitaires et donner une URL pour la Doc. java : dans le Menu 'Outils', choisir l'item 'Préférences', sélectionner l'onglet 'Divers' puis donner une URL et cocher comme ci dessous

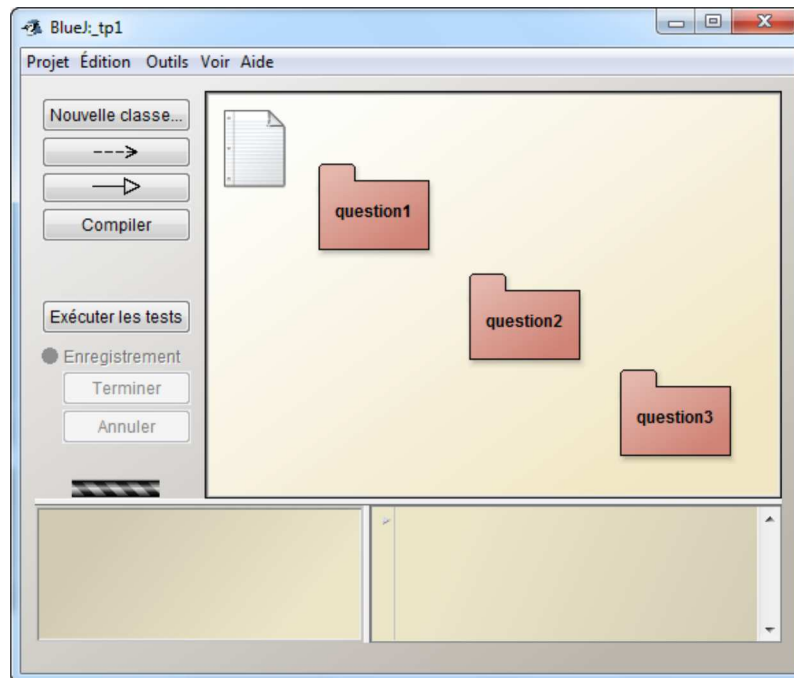


introduction

**Utilisation de BlueJ et soumission de vos travaux au CNAM**

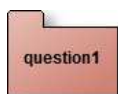
1. Télécharger l'archive 'tp1.jar', dans le répertoire de votre choix, par exemple 'E:/tp\_java/'
2. Lancer Bluej, puis dans le Menu 'Projet', choisir l'item 'Ouvrir non-bluej...', sélectionnez cette archive 'tp1.jar'

Vous obtenez :



Les 3 questions correspondent aux 3 paquetages

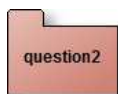
- Votre projet se trouve maintenant dans le répertoire : 'E:/tp\_java/tp1/'
- L'énoncé du TP/TD ('tp1') est dans ce répertoire en 'E:/tp\_java/tp1/tp1.html'.



## Introduction

Champs, constructeurs, méthodes et instances d'une classe (introduction extraite du livre Objects First de M .Kölling, [chapitre 1 de Objects-First](#)).

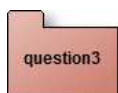
1. Allez dans le paquetage question1
2. Puis créer plusieurs instances,
3. Déclenchez les méthodes d'instance,
4. Vérifier les champs d'instance des classes "Circle, Square et Triangle"
5. Une première [documentation](#) , ( générée par Bluej Menu Outils)



## Introduction suite

1. Créer une instance de la classe Picture, Modifier le source "Picture.java" afin que le soleil soit bleu,
2. Ajouter une nouvelle donnée d'instance : un nouveau soleil (jaune...),
3. Ajouter une nouvelle méthode à la classe Picture permettant au soleil ( bleu) de se coucher (si la terre est fixe...), voir la méthode slowMoveVertical de la classe "Circle"

Répondre aux autres exercices du [chapitre 1 de Objects-First](#)



## Introduction, tests, mise en oeuvre

1. Le paquetage question3 contient la classe AuditeurCnam fondée sur le squelette de programme suivant :

```
public class AuditeurCnam {
    /** l'attribut nom de chaque auditeur. */
    private String nom;
    /** l'attribut prenom de chaque auditeur. */
    private String prenom;
```

```

/** l'attribut matricule de chaque auditeur. */
private String matricule;

/**
 * "Création", le constructeur d'un auditeur avec son nom, son prénom et son
 * matricule.
 *
 * @param nom
 *         le nom de l'auditeur
 * @param prenom
 *         son prénom
 * @param matricule
 *         sur la carte d'inscription, près de la photo
 */
public AuditeurCNAM(String nom, String prenom, String matricule) {
    this.nom = nom;
    this.prenom = prenom;
    this.matricule = matricule;
}

/**
 * le login au Cnam : 6 premières lettres du nom suivies de la première
 * lettre du prénom séparées de '_' . le login retourné est en minuscules,
 * le trait d'union, ou spéciaux <i>(pour unix)</i> sont remplacés par des
 * '_' pas de caractères accentués pour le login voir les classes
 * prédéfinies, java.lang.String : les méthodes replaceAll, toLowerCase et
 * substring java.lang.Math : la méthode min<br>
 * <b>BlueJ : Menu Aide</b>
 *
 * @return le login du Cnam simplifié, sans les adaptations dues aux
 *         homonymes...
 */
public String login() {
    return ""; // à compléter
}

/**
 * Lecture du nom de l'auditeur.
 *
 * @return son nom
 */
public String nom() {
    return null; // à compléter
}

/**
 * Lecture du prénom de l'auditeur.
 *
 * @return son prénom
 */
public String prenom() {
    return null; // à compléter
}

/**
 * Lecture du matricule de l'auditeur.
 *
 * @return son matricule
 */
public String matricule() {
    return null; // à compléter
}

/**
 * méthode toString ( méthode redéfinie de la classe Object).
 *
 * @return la concaténation du nom, du prénom et du login, selon cette
 *         syntaxe
 *         <code>nom() + " " + prenom() + " login : " + login()</code>
 */
@Override
public String toString() {
    return nom() + " " + prenom() + " login : " + login();
}
}

```

2. Lire la [documentation](#) de cette classe
3. Complétez cette classe
4. Créer plusieurs instances (clic droit sur la classe), vérifiez la valeur des données d'instances avec l'inspecteur d'objets
5. Exécutez les tests unitaires proposés, ceux-ci sont minimaux
6. Ajoutez vos propres tests, voir le tutoriel "[Unit Testing in BlueJ](#)"

7. Effectuez une évaluation de votre classe avec [JNEWS](#) installé au Cnam :

Menu Outils, item Submit voir le mode d'emploi en <http://jfod.cnam.fr/progAvancee/>  
 Entrez vos Nom, et Matricule si vous n'êtes pas encore inscrit, vérifiez à l'aide de [cette page](#)  
 à ce lien, la liste provisoire des [inscrits en NFP121](#)

