

# Projet tutoré du semestre 2

Isabelle Robba – 27 mai 2015

Le projet tutoré vous donne l'occasion de mettre en application les connaissances acquises dans trois des modules informatiques du second semestre : les bases de la conception orientée objet, les bases de la programmation orientée objet et l'introduction aux interfaces homme-machine. Il est réalisé en binôme, pour vous permettre d'acquérir une première expérience de travail en équipe. Il est important que les 2 membres du binôme participent de façon équilibrée à la mise en œuvre du projet : les tâches de conception, de réalisation et de documentation doivent être partagées de la façon la plus équitable possible.

## 1 Présentation

L'application à réaliser doit permettre la mise en place du programme des festivals de musique de l'été 2015. Plusieurs genres de musique seront représentés : musique classique, jazz et blues, rock, musiques du monde... L'application doit contenir plusieurs interfaces correspondant à ses différentes fonctionnalités. Il est demandé une interface contenant un formulaire pour saisir les différents festivals de musique composant le programme de l'été, une interface affichant de façon détaillée les festivals et une dernière interface permettant de réserver des places.

## 2 Conception

Pour la conception, vous devez appliquer l'approche orientée objet (cf cours de BCOO). Le dossier de conception comportera une analyse de la structure du système ainsi que de son comportement. Vous proposerez donc des scénarios d'utilisation de votre logiciel. Le dossier reposera sur une modélisation en UML (diagramme de classes, diagramme d'interactions...). Vous écrirez également un dossier de tests.

## 3 Réalisation

### 3.1 Les fonctionnalités de l'application

Les fonctionnalités suivantes sont à réaliser :

- la saisie à l'aide d'un formulaire des informations concernant un festival,
- l'affichage de chaque festival sous forme détaillée, cet affichage est réalisé à l'aide d'une table à plusieurs colonnes, chaque colonne contenant les festivals de musique d'un même genre (musique classique, jazz et blues, rock ...)
- la gestion rudimentaire des réservations : dans la copie d'écran de la figure 1, chaque bouton de la grille contient l'affiche, le nom, les dates et le genre de musique et le lieu d'un festival. Quand un festival sera terminé ou que toutes ses réservations seront complètes, le bouton correspondant sera désactivé. En cliquant sur un bouton actif, l'utilisateur aura accès aux différentes dates auxquelles il restera encore des places pour ce festival ; en choisissant une date, il effectuera une réservation (le nombre de réservations pour ce spectacle sera simplement décrémenté). La gestion est rudimentaire en ce sens qu'on ne demande pas de gérer les différentes catégories de place, les différents prix proposés, les utilisateurs, on suppose en outre qu'à chaque date correspond un seul concert, donc une seule réservation...
- la sauvegarde sur fichier des données entrées par l'utilisateur, les données concernées sont d'une part les informations entrées dans le formulaire de saisie, d'autre part les informations liées aux réservations

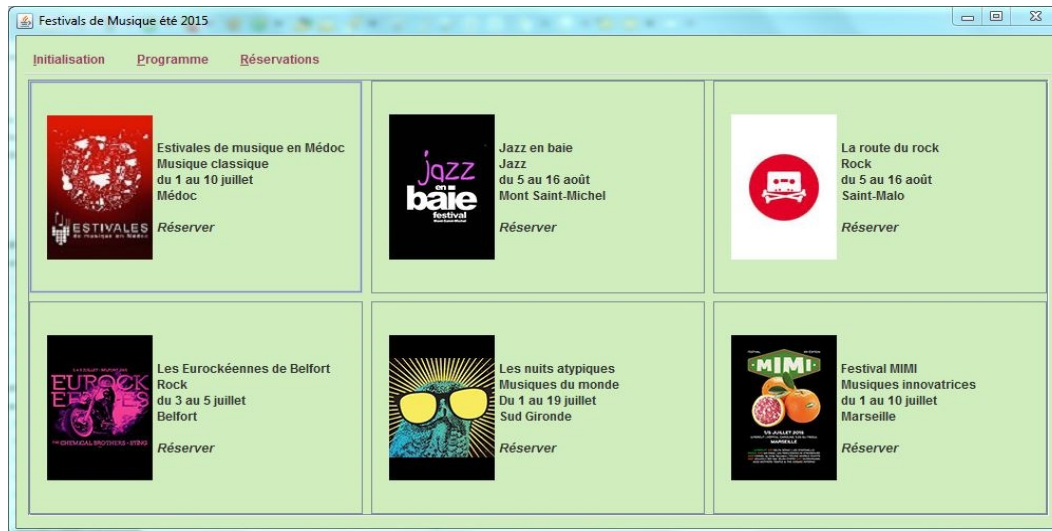


FIGURE 1 – L’interface de réservation

## 3.2 Le modèle de l’application

Les classes nécessaires pour composer l’application :

- la classe *Festival* qui contient toutes les informations pour décrire un festival, entre autres : la date de début et celle de fin du festival, le lieu, le genre de musique, éventuellement un genre plus précis (rock alternatif, musiques innovatrices...) une description du festival, le nombre de places pouvant être réservées, le nom du fichier contenant l’affiche... liste non exhaustive,
- la classe *Programme* qui contient dans une structure de données de votre choix tous les festivals de l’été, il sera opportun de choisir une structure de données dans laquelle les festivals sont regroupés par genre de musique

À ces classes vous pourrez ajouter toutes les classes que vous jugerez pertinentes.

## 3.3 Les interfaces

À l’aide d’un menu, l’utilisateur pourra visualiser les différentes interfaces de l’application, celle-ci correspondant aux trois premières fonctionnalités de l’application :

- l’interface de saisie
- l’interface d’affichage des festival dans une table où chaque colonne correspond à un même genre de musique
- l’interface de réservation, chaque festival y est représenté par un bouton sur lequel on peut cliquer pour réserver une date. Une solution basique sera ici d’ouvrir un objet de la classe *JOptionPane* contenant dans une *JComboBox* les dates possibles ; une solution plus avancée pourra être d’utiliser un objet de la classe *JDialog* qui permet de proposer une interface plus complète de réservation. Un exemple de code est donné en annexe, section 7.

## 4 Le rapport

Le rapport contient plusieurs parties :

1. le dossier de conception générale : il recueille le diagramme des classes, les scénarios, avec pour chacune des parties les explications nécessaires et pas seulement les diagrammes.
2. le dossier de conception détaillée : il précise la conception générale à un niveau d’abstraction plus bas (proche du code), les structures de données y seront détaillées, sa lecture doit permettre au développeur de modifier ou compléter l’application.

3. le dossier de tests : il contient les tests unitaires (tests boîtes noires), leur conception, leur préparation et leur analyse
4. le manuel utilisateur : il présente l'application et ses fonctionnalités à l'utilisateur, il peut contenir des copies d'écran pour illustrer les explications.

Le rapport sera déposé, **au plus tard le lundi 8 juin à 9 heures**, sous forme électronique, sur ecampus2 dans l'espace Projet tutoré du semestre 2. Il sera accompagné du code de l'application et de la javadoc associée à ce code. Le document déposé doit avoir pour intitulé les noms des 2 membres du binôme.

## 5 La soutenance

La soutenance aura lieu **le lundi 8 juin**, elle permet aux 2 membres du binôme de présenter l'application sous la forme d'une démonstration et de répondre aux questions du jury.

## 6 Remarques et conseils

1. Ne sous-estimez pas la quantité de travail à effectuer : ne codez pas directement, commencez par la conception ; la phase de programmation est toujours plus longue que l'on ne croit ; la rédaction du rapport a souvent tendance à être bâclée, mais notez bien que la réalisation et le rapport auront le même coefficient dans la notation. Enfin, en ce qui concerne la documentation, écrivez la au fur et à mesure que vous codez et non une fois que tout le code est écrit.
2. L'utilisation de GIT peut être envisagée, mais elle n'est pas obligatoire.
3. Dans la mesure de leurs compétences, les 2 membres du binôme doivent participer à toutes les tâches et ce de façon équilibrée.

## 7 Annexe

Le code qui suit permet d'ouvrir un *JDialog* dans lequel on pourra remplacer la *JLabel* par une interface complète de réservation. Notez l'utilisation des balises html nécessaire pour afficher le texte de la *JLabel* sur plusieurs lignes.

```
public class PanelTestJDialog extends JPanel implements ActionListener {

    public PanelTestJDialog() {
        JButton bouton = new JButton ("Cliquer pour ouvrir un JDialog");
        bouton.addActionListener(this);
        this.add(bouton);
    }

    public void actionPerformed(ActionEvent evt) {
        JDialog dial = new JDialog ( );
        dial.setModal(true);
        Point positionSouris = getMousePosition();
        dial.setSize(370,240);
        dial.setLocation(positionSouris );
        dial.setTitle("La réservation");
        dial.setContentPane(new JLabel  ("<html> Placer ici au lieu d'une JLabel <br/>" +
            "un panel pour effectuer une réservation</html>"));
        dial.setVisible (true);
    }

}
```