

Examen de Java

Nom(s) :

Groupe :

Date :

Durée 1h30, tous documents personnels autorisés, calculatrices et ordinateurs interdits. Le sujet comporte deux parties indépendantes, les réponses seront écrites sur le sujet dans les emplacements prévus à cet effet.

1 Exercice 1 : Structure du code

On considère le code source incomplet donné ci-après. Plusieurs questions sont posées dans les commentaires de ce code. Répondez-y dans les espaces prévus à la suite du programme.

```
class QuestionsFaciles {
    ...

    public static void main(String[] args) {
        ...

        while (current < args.length) {
            /*
             * Question 1 : qu'est-ce que "args" ? Qu'est-ce que
             * "length" ? Qu'est-ce que "args.length" ?
             */
            if (args[current].equals("-help"))
                help();
            else if (args[current].equals("-port")) {
                port_number = Integer.parseInt(args[current + 1]);
                /*
                 * Question 2 : qu'est-ce que "Integer" ? et
                 * "Integer.parseInt(...)" ?
                 *
                 * Question 3 : quelle est l'utilité de
                 * l'instruction Integer.parseInt(...)?
                 *
                 * Question 4 : Sachant que cette instruction
                 * retourne une Exception, quel code devrait-on
                 * rajouter pour compléter ce source. Utilisez
                 * notamment la méthode printerr(...) définie
                 * plus bas
                 */
                current += 2;
            } else
                throw new Exception(args[current]);
            /*
             * Question 5 : Quel est le rôle de l'instruction
             * "throw" et quel est l'effet de l'instruction
             * "new Exception (...);"
             */
        }
    }
}
```

```
private static void printerr(String message, Exception e) {
    /*
     * Question 6 : que signifie la déclaration
     * "private static void" pour cette méthode ?
     */
    System.err.println("erreur: " + message);
    /*
     * Question 7 : Explicitez chacune des entités composant
     * l'instruction : "System" "err" "println(...)"
     */
    e.printStackTrace();
    System.exit(1);
}
...
}
```

Réponse à la question 1 :

Réponse à la question 2 :

Réponse à la question 3 :

Réponse à la question 4 :

Réponse à la question 5 :

Réponse à la question 6 :

Réponse à la question 7 :

2 Exercice 2 : Jackpot

Le Jackpot est un jeu de hasard : dans une machine on insère un jeton, on actionne un levier et trois roues sur lesquelles sont affichées des images tournent. Si une fois arrêtées les trois roues n'affichent pas la même chose, on a perdu. Si elles affichent toutes le chiffre 7, on gagne 500 jetons. Dans tous les autres cas où elles affichent la même chose, on gagne 20 jetons.

Dans cet exercice, on vous demande de recréer ce jeu graphiquement : les roues sont représentées par des boutons (le label du bouton indique « ? » avant de jouer, puis un chiffre entre 0 et 9 une fois la machine actionnée). Un label donne le nombre de jetons restants du joueur. Un bouton permet au joueur de miser un jeton (et de remettre les roues à « ? ») et enfin un autre permet d'actionner le Jackpot.

Lorsque le joueur aura misé un jeton et actionné le Jackpot, les roues afficheront aléatoirement leur résultat. Si le joueur a gagné quelque chose, son nombre de jetons sera augmenté en conséquence. Le joueur aura 100 jetons en début de partie.

Décrivez la décomposition en éléments graphiques et *layout managers* de votre application (par des dessins au besoin).

Indications avant de donner le code de l'application :

- Pour modifier le texte d'un label `label`, ou d'un bouton `bouton`, on peut utiliser les méthodes `setText` des classes `JLabel` et `JButton`. Par exemple : `label.setText("Nouveau label");` ou `bouton.setText("Nouveau bouton");`
- Pour générer un nombre entier aléatoirement, on peut utiliser la méthode non statique `nextInt()` de la classe `Random` qui renvoie un nombre entier (type `int`) aléatoire.
- Le programme pouvant être long, **veuillez à ne pas écrire du code redondant**, indiquez simplement si le code est semblable pour telle ou telle partie, voire telle ou telle ligne de code, par exemple :

```
getContentPane().add(roue1) ;  
-----roue2---  
-----roue3---
```

Implémentez ce jeu en JAVA en pensant à prévenir toute situation anormale avec des exceptions (plus de jetons, lancement du Jackpot sans donner de jeton etc.).



On souhaite qu'après chaque lancement du jackpot, le nombre de jetons soit conservé dans un fichier. Que faut-il rajouter et où faut-il le rajouter pour cela ?

En imaginant que vous avez eu le temps de réaliser l'application **complète**, est-il possible de faire de ce jackpot une applet de manière à y jouer depuis un navigateur web ? Si non, pourquoi, si oui, comment (dans les grandes lignes) ?