

Java DUT 1 Feuille TD7
Université Paris-Est Marne-la-Vallée

Exercice 1.—

Flots d'octets en lecture

- a) Tester le programme suivant. Que fait-il?

```
import java.io.*;

public class ReadTest {
    public static void main(String[] args){
        try {
            int i = System.in.read();
            System.out.println(i);
        } catch (IOException e) {};
    }
}
```

- b) Modifiez le pour qu'il lise une ligne de caractères entrée au clavier, caractère par caractère, et affiche ensuite le code ASCII de chaque caractère de la ligne. La ligne se termine par le caractère de passage à la ligne \n.

Exercice 2.— Lecture d'entiers avec la classe `Scanner`.

- a) Que fait la méthode `main` de la classe suivante?

```
import java.util.*;

class IntRead {
    public static void main (String[] args) {
        Scanner sc = new Scanner(System.in);
        int i;
        if (sc.hasNext()){
            i = sc.nextInt();
            System.out.println(i);
        }
        sc.close();
    }
}
```

- b) Modifier cette méthode pour pouvoir lire plusieurs entiers entrés au clavier.

Exercice 3.— Lecture de mots avec la classe `Scanner`.

- a) Ecrire une méthode `main` dans une classe `StringRead` qui permet de lire des mots entrés au clavier et de les afficher. On complètera le code suivant.

```
import java.util.*;

class StringRead {
    public static void main (String[] args) {
        Scanner sc = new Scanner(System.in);
        String s;
        .....
        System.out.println(s);
    }
}
```

```

        sc.close();
    }
}

```

- b) Modifier cette méthode pour afficher les mots en les mettant en majuscule.

Exercice 4.—

Lire un texte sur l'entrée standard.

- a) Que fait la méthode `main` de la classe suivante? La méthode `String readLine()` de la classe `BufferedReader` retourne la ligne de caractères suivante dans le flot (le caractère de fin de ligne non compris) sous forme de `String`.

```

class Test {
    public static void main(String[] args) throws IOException {
        BufferedReader in = null;
        try {
            in = new BufferedReader(
                new InputStreamReader(System.in));
            String s;
            if ((s = in.readLine()) != null) {
                System.out.println(s);
            }
        } catch (IOException e) {
        } finally {
            if (in != null) in.close();
        }
    }
}

```

- b) Le `throws` dans l'entête de la méthode `main` est-il nécessaire?
- c) Pourquoi la ligne `BufferedReader in = null;` est-elle en dehors du bloc `try-catch`?
- d) Modifier le programme pour lire toutes les lignes et les afficher.
- e) Modifier le programme pour lire toutes les lignes et les afficher en mettant les mots en majuscule.

Exercice 5.—

Ecrire des classes et méthodes pour lire un fichier texte et afficher ensuite un histogramme du texte : la liste des mots présents dans le texte et pour chacun d'eux, leur nombre d'occurrences.

On pourra traiter l'entrée standard et utiliser les redirection pour traiter un fichier.