

Enseignement secondaire classique
Section informatique et communication
Examen 1CI/1CLI - 2024/2025

Liste de codes usuels

Écriture dans un fichier texte

```
try (PrintWriter out = new PrintWriter(new FileWriter(fileName, StandardCharsets.UTF_8))) {  
    out.println("some text");  
}
```

Lecture depuis un fichier texte

```
try (BufferedReader in = new BufferedReader(new FileReader(fileName, StandardCharsets.UTF_8))) {  
    String line;  
    while ((line = in.readLine()) != null) {  
        // do something with line  
    }  
}
```

Conversion d'un objet en JSON (à l'aide de la librairie GSON)

```
Gson gson = new GsonBuilder().create();  
String json = gson.toJson(someObject);
```

Conversion d'un objet en JSON avec adaptateur (à l'aide de la librairie GSON)

```
Gson gson = new GsonBuilder()  
    .registerTypeAdapter(Color.class, new ColorAdapter())  
    .create();  
String json = gson.toJson(someObject);
```

Conversion d'un texte JSON en objet (à l'aide de la librairie GSON)

```
Gson gson = new GsonBuilder().create();  
Foo foo = gson.fromJson(json, Foo.class);
```

Conversion d'un texte JSON en objet avec adaptateur (à l'aide de la librairie GSON)

```
Gson gson = new GsonBuilder()  
    .registerTypeAdapter(Color.class, new ColorAdapter())  
    .create();  
Foo foo = gson.fromJson(json, Foo.class);
```

Parcours d'une table associative

```
Map<String, Integer> m = ...;  
for (Map.Entry<String, Integer> e : m.entrySet()) {  
    System.out.println(e.getKey() + "->" + e.getValue());  
}
```

Détermination de la largeur et de la hauteur d'un texte dessiné sur un canevas g

```
int stringWidth = g.getFontMetrics().stringWidth(texte);  
int stringHeight = g.getFontMetrics().getHeight();
```

Extraction des fichiers d'un répertoire

```
File directory = new File("c:\some\directory");  
File[] entries = directory.listFiles();
```

Connexion à une base de données

```
import java.sql.*;  
  
try {  
    String server = "localhost";  
    int port = 3306;  
    String username = "alice";  
    String password = "security";  
    String database = "school";  
  
    Class.forName("com.mysql.cj.jdbc.Driver");  
  
    String url = "jdbc:mysql://" + server + ":" + port + "/" +  
                database + "?user=" + username + "&password=" + password;  
  
    try (Connection connect = DriverManager.getConnection(url)) {  
        try (Statement statement = connect.createStatement()) {  
            ResultSet resultSet = statement.executeQuery("SELECT * FROM people");  
  
            while (resultSet.next()) {  
                String name = resultSet.getString("name");  
                System.out.println(name);  
            }  
        } catch (SQLException ex) {  
            System.out.println("SQLException: " + ex.getMessage());  
            System.out.println("SQLState: " + ex.getSQLState());  
            System.out.println("VendorError: " + ex.getErrorCode());  
        }  
    } catch (SQLException ex) {  
        System.out.println("SQLException: " + ex.getMessage());  
        System.out.println("SQLState: " + ex.getSQLState());  
        System.out.println("VendorError: " + ex.getErrorCode());  
    }  
} catch (ClassNotFoundException ex) {  
    Logger.getLogger(DatabaseTest.class.getName()).log(Level.SEVERE, null, ex);  
}
```