

Nom, prénom :

Code permanent :

Répondez directement sur le questionnaire. Aucune documentation n'est permise. Chaque réponse doit tenir dans les lignes tracées à cet effet.

Question #1 – 10%

Décrivez 3 pratiques ou techniques visant à améliorer la maintenabilité du code source d'un projet.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Question #2 – 5%

Quel outil recommanderiez-vous pour la rédaction de la documentation utilisateur d'un logiciel et pourquoi?

.....

.....

.....

.....

Question #3 – 15%

Décrivez 5 étapes de refactoring que vous pourriez effectuer dans le code source fourni en annexe.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Question #4 – 5%

Qu'est-ce qu'une analyse organique selon le modèle de développement en cascade?

.....

.....

.....

.....

.....

Question #5 – 5%

Un gestionnaire de sources est considéré comme un outil collaboratif. Pourquoi?

.....
.....
.....
.....

Question #6 – 5%

Comment un gestionnaire de sources permet-il de favoriser l'agilité d'une équipe de développement?

.....
.....
.....
.....

Question #7 – 5%

Qu'est-ce que le manifeste Agile?

.....
.....
.....
.....

Question #8 – 5%

Avec le modèle de développement itératif, à quelle fréquence une rétroaction du client est-elle sollicitée?

.....

.....

.....

.....

Question #9 – 5%

La casse des identifiants est toujours précisée dans une norme de codification. Pourquoi?

.....

.....

.....

.....

Question #10 – 5%

Dans le chapitre 3 du livre Coder proprement, l'auteur recommande de n'avoir qu'un seul niveau d'abstraction par fonction. Qu'est-ce que cela signifie?

.....

.....

.....

.....

Question #11 – 5%

Dans le chapitre 3 du livre Coder proprement, quelle est la position de l'auteur face aux arguments d'une fonction?

.....

.....

.....

.....

Question #12 – 5%

Dans le chapitre 4 du livre Coder proprement, l'auteur indique que seuls les bons commentaires devraient se retrouver dans le code source. Selon lui, qu'est-ce qu'un bon commentaire?

.....

.....

.....

.....

Question #13 – 5%

Dans le chapitre 7 du livre Coder proprement, quelle est la position de l'auteur au sujet de l'utilisation des codes de retour pour la gestion des erreurs?

.....

.....

.....

.....

Question #14 – 10%

Comment le modèle de développement itératif permet-il à une équipe de développement d'améliorer sa productivité au fur et à mesure que le projet avance?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Question #15 – 10%

Si nous développons un logiciel selon le modèle en cascade et que nous sommes rendu à l'étape de la réalisation, que devons-nous faire pour gérer un changement dans les besoins du client?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Annexe – Code source question #3

```

/**
 * Generates the getAll method for the DAO.
 * @param writer The file writer.
 * @param entity The entity of the DAO.
 * @throws IOException From the file writer.
 */
public static void provideDAOGetAllMethod(final FileWriterWrapper writer,
                                         final Entity entity)
                                         throws IOException {
    writer.write("    public ArrayList<" + entity.getJavaName()
                + "> getAll" + entity.getJavaName()
                + "s() throws DAOException {\n");
    writer.write("        try {\n");

    writer.write("            PreparedStatement query = "
                + "connection.prepareStatement(\"select * from "
                + entity.getSqlName() + "\");\n\n");

    writer.write("            ResultSet result = query.executeQuery();\n");
    writer.write("            result.beforeFirst();\n");

    writer.write("            ArrayList<" + entity.getJavaName()
                + "> list = new ArrayList<" + entity.getJavaName()
                + ">();\n\n");

    writer.write("            while (result.next()) {\n");

    String variableName = entity.getJavaName().toLowerCase();
    writer.write("                " + entity.getJavaName()
                + " " + variableName + " = new " + entity.getJavaName()
                + "();\n");
    writer.write("                " + variableName
                + ".setId(result.getInt(\"" + entity.getSqlName()
                + "_ID\")); \n\n");

    for (Field field : entity.getFields()) {
        writer.write("                    " + variableName + "."
                    + field.getSetterName() + "(result."
                    + mapResultSetGetter(field.getOriginalDataType())
                    + "(" + field.getSqlName() + "));\n");
    }
    writer.write("\n");

    for (Relation manyToOne : entity.getAllManyToOneRelations()) {
        Entity link = manyToOne.getEntity();
        writer.write("                " + variableName + ".set"
                    + link.getJavaName() + "Id(result.getInt(\""
                    + link.getSqlName() + "_ID\")); \n");
    }
    writer.write("\n");

    for (Relation many : entity.getAllMANYRelations()) {
        Entity link = many.getEntity();
        String statementName = link.getJavaName().toLowerCase() + "s";
        String linkTable = link.getSqlName();
        if (many.isManyToMany()) {

```

INF2015 – Développement de logiciels dans un environnement Agile

```
        linkTable = many.getNameOfLinkTable();
    }

    writer.write("                PreparedStatement " + statementName
        + " = connection.prepareStatement(\"select \"
        + link.getSqlName() + \"_ID from \" + linkTable
        + \" where \" + entity.getSqlName() + \"_ID=?\");\n");
    writer.write("                " + statementName + ".setInt(1, \"
        + variableName + \".getId().intValue());\n");

    String resultName = statementName + "Result";
    writer.write("                ResultSet " + resultName + " = \"
        + statementName + \".executeQuery();\n");
    writer.write("                " + resultName + ".beforeFirst();\n");
    writer.write("                while (" + resultName
        + ".next()) {\n");
    writer.write("                    " + variableName + ".add\"
        + link.getJavaName() + "(" + resultName + ".getInt(\"\"
        + link.getSqlName() + \"_ID\")\");\n");
    writer.write("                }\n\n");
}

writer.write("                list.add(" + variableName + ");\n");
writer.write("                }\n\n");
writer.write("                return list;\n");

writer.write("            } catch (SQLException e) {\n");
writer.write("                throw new DAOException(\"Unable to perform \"
        + \"query on database.\", e);\n");
writer.write("            }\n");
writer.write("        }\n\n");
}
```