RECRUITMENT TEST (JAVA DEVELOPER)

Instruction: Please answer the following questions to the best of your ability and return your solutions to X-Formation for review.

Note the following considerations before you start taking the test:

- The usage of Maven in Java projects would be considered a plus.
- Hosting your project in a public repository (e.g. GitHub) would be an advantage.

TASK DESCRIPTIONS

1. Food ordering system

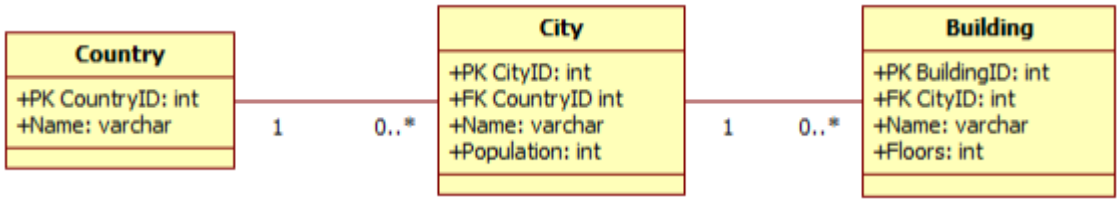
Imagine you have to design an application for a restaurant ordering system. Please follow the requirements listed below:

- Create only command-line interface (no UI is required)
- It's possible to order lunch or drink via command-line interface
- There are three cuisines available to choose from (Polish, Mexican, Italian)
- Lunch consists of the main course and dessert
- When ordering a drink, the customer can additionally ask for ice cubes or/and lemon
- We must have getPrice() and getName() methods for each meal and drink
- The solution you come up with should be extendable (it should be possible to add new cuisines/dishes in the future)

Present your proposal of Java implementation for this application.

2. SQL

Consider the following physical data model:



Write two SQL queries that:

1. Select countries where a total number of inhabitants (population) in all cities is greater than 400.
2. Select names of the countries that have no buildings at all.

Please use the schema named "XF" & same table+fields naming as shown in the diagram.

3. MineSweeper

Please provide the implementation of the following interface :

```
public interface MineSweeper {

    /**
     * Initialises the field
        *
     * A mine-field of N x M squares is represented by N lines of exactly M characters each.
     * The character '*' represents a mine * and the character '.' represents no-mine.
     * Lines are separated by '\n'
     * <p/>
     *  * Example mine-field string (as input to setMineField()): "*...\n..*.\n...."
     * (a 3 x 4 mine-field of 12 squares, 2 of which are mines)
     *
```

```
    * @param mineField string containing the mines
    * @throws IllegalArgumentException if mineField is not properly formatted
    */
   void setMineField(String mineField) throws IllegalArgumentException;

   /**
    * Produces a hint-field of identical dimensions as the mineFiled() where each
        * square is a * for a mine or the number of adjacent mine-squares if the square does
not contain a mine.
        * <p/>
        * Example hint-field (for the above input): "*211\n12*1\n0111"
    *
    *  @return a string representation of the hint-field
    *  @throws IllegalStateException if the mine-field has not been initialised (i.e.
setMineField() has not been called)
    */
   String getHintField() throws IllegalStateException;

}
```