CASE SMARTBBQ  
With the current corona crisis, a well-known BBQ manufacturer sees a gap in the market. To prevent many people from gathering around a BBQ, they have an idea to create a SmartBBQ™. The following paragraphs list the main features of the program.

SMARTBBQ™  
The SmartBBQ™ can be set to an exact temperature in whole degrees Celsius, with a maximum of 250 degrees Celsius.

By default, the BBQ is off. A maximum of six food items can be placed on the SmartBBQ™. This is predetermined.

When one turns on the SmartBBQ™ with the turnOn function, the temperature should be given along with it. The SmartBBQ™ must also include a preheat function, which raises the temperature to the desired level in increments of 5°C per second. This preheating process must utilize a Timer and TimerTask to simulate the heating effect dynamically and stop once the set temperature is reached.

MEAT  
Different types of meat can be put on the barbecue. For the barbecue, it is

important to know the following characteristics:

• Current cooking percentage **(0 - ∞)**

• Current tanning percentage **(0 - ∞)**

• Meat origin: Kip/Koe/Varken

Meat is a subclass of the abstract class FoodItem and must override the grill method inherited from the superclass. Meat also contains a toString() method that returns the name of the meat.

VEGETABLES  
Not only meat can go on the barbecue, but also other foods, such as vegetables. Vegetables are also a subclass of FoodItem.

For the barbecue, it is important to know the following characteristics:

• Current tanning percentage **(0 - ∞)**

• Moisture percentage **(100% - 0 %)**

Vegetables must override the grill method inherited from FoodItem. Vegetables contain a toString() method that returns the name of the vegetable.

FOODITEM  
The abstract superclass FoodItem contains the following shared characteristics: a current temperature T, which increases by 0.5 degrees per second during grilling, and a grill method, which is abstract and must be implemented by all subclasses. The grill method depends on two main elements: its temperature and grill duration in seconds. It also calculates specific properties, such as cooking, tanning, and moisture changes for each food type based on the temperature of the SmartBBQ™. The method will not return any value after running.

| **FOOD** | **COOKED INCREASE/SECOND** | **TAN INCREASE/SECOND** | **MOISTURE DECREASE/SECOND** | **MEAT ORIGIN** |
| --- | --- | --- | --- | --- |
| HAMBURGER | (1 / (100 ∗ π)) ∗ T | 0.06 ∗ (T ∗ 100) | n.v.t. | Koe |
| SAUSAGE | (1 / (80 ∗ π)) ∗ T | 0.05 ∗ (T / 100) | n.v.t. | Varken |
| BELL PEPPER | n.v.t. | 0.03 ∗ T | 0.1 ∗ T | n.v.t. |
| CORN | n.v.t. | 0.08 ∗ T | 0.05 ∗ T | n.v.t. |

FRIDGE  
There is one fridge (singleton) in which food is placed and taken out. The fridge can hold an infinite amount of food. The user can only choose whether he wants meat or vegetables. Not which specific kind. The user then gets the first available food item of the type he chose.

If no more food is available of the selected type, an exception is raised: NoMoreFoodException.

The fridge temperature can be set, but defaults to 8 degrees Celsius.

THERMOMETER  
The thermometer can be used to query the temperature of various elements. This includes the fridge, the BBQ, and food items. The thermometer class contains a single measureTemperature method that takes any object implementing the Temperature interface. The Temperature interface must be implemented by the fridge, SmartBBQ™, and FoodItem classes. The method should return a string with the text: “I measured a temperature of xx degrees Celsius,” along with a classification of the temperature. Anything below 10 degrees is cold. Between 10 and 70 degrees is medium. Anything above that is hot.

ASSIGNMENT 1  
Build the application based on the given class diagram and the text above.

ASSIGNMENT 2  
The turnOn feature in the SmartBBQ™ also starts a preheating timer that increases the temperature dynamically in 5°C increments until the desired temperature is reached. Use Timer and TimerTask for this feature and ensure that the user is notified when the BBQ is ready.

ASSIGNMENT 3  
The SmartBBQ™ must support dynamic grilling sessions. Implement a startGrillSession method in the SmartBBQ™ that periodically (every second) triggers the grill method for all food items currently on the BBQ. This session must use Timer and TimerTask and must stop automatically when all food items have reached a fully cooked state (100% cooking percentage). Add functionality to notify the user when the grilling session is complete.

ASSIGNMENT 4  
Implement the FridgeIsEmptyException and throw it as soon as an attempt is made to take food from an empty fridge. Add logic to check the temperature of the fridge using the thermometer and ensure that any food retrieved from a fridge that is too warm (above 10°C) is marked as spoiled and cannot be placed on the SmartBBQ™.

ASSIGNMENT 5  
a) Write a meaningful Unit Test for the getNextMeat method to ensure it always retrieves the correct type of food.  
b) Write a Unit Test to simulate grilling vegetables and verify that vegetables with a moisture percentage below 5% are too dry to consume.

ASSIGNMENT 6  
Create a Main class with a static main method that:  
Makes a SmartBBQ™. Puts three corn cobs and two bell peppers in the fridge. Puts two hamburgers and three sausages in the fridge. Puts three pieces of meat and two vegetables on the SmartBBQ™. Turns on the SmartBBQ™ with a temperature of 180 degrees Celsius, preheats it to the desired temperature, and starts the grill session. Measures the temperature of one piece of meat using the thermometer. Increases the temperature of the SmartBBQ™ to 200 degrees Celsius during the session and measures the temperature of the same piece of meat again. Simulates the grilling session until all food is cooked and notifies the user when complete.