Eli Zupke

December 4, 2017

The GitHub folder can be found at https://github.com/Trainzack/CS240/tree/master/Final%20Part%201.

Contents

1 src/Inventory.java

```java
/**
 * Simulates a simplified model of an In-N-Out.
 * @author eli
 *
 */

public class Inventory {

    // What day of the month it is. Starts at DEC 1st, or 301.
    private static int day = 301;

    // What is the last day to simulate (DEC 31st)
    private static final int last_date = 331;

    // This array contains the six menu items that customers can order.
    ListInterface[] menuItems = new ListInterface[6];


    /**
     * Intializes all of the important variables
     */
    private static void init() {

    }


    /**
     * Simulates a single day.
     */
    private static void runDay() {

    }


}
```

2 src/FoodType.java

```java
/**
 * Represents the type of a single food item. Each type has an associated name, and expiry time (in days)
 */


public enum FoodType {
    Bun ("Bun", 5),
    Patty ("Patty", 4),
    Lettuce ("Lettuce", 3),
    Tomato ("Tomato", 3),
    Onion ("Onion", 5),
    Cheese ("Cheese", 2)
    ;
    private final int expirationTime;
    private final String name;

    private FoodType(String name, int expirationTime) {
        this.name = name;
        this.expirationTime = expirationTime;
    }

    @Override
    public String toString() {
        return name;
    }

    /**
     * Returns the amount of days that this type of food item lasts for
     * @return An int represnting the number of days that this food item lasts for
     */
    public int getExpirationTime() {
        return expirationTime;
    }
}
```

3 src/FoodItem.java

```java
/**
 * Represents a single, actual ingredient that exists. It has an expiry date and everything!
 * @author eli
 *
 */
public class FoodItem {

    // The day that this item will expire.
    private int expiryDate;

    // What type of food item this
    private FoodType type;

    /**
     * Creates a new food item of the given type, and sets the expiration date accordingly
     * @param curDate The current day, as an int.
     * @param type What type of food item this is.
     */
    public FoodItem(int curDate, FoodType type) {
        super();
        this.type = type;
        this.expiryDate = curDate + type.getExpirationTime();
    }

    /**
     * Checks whether this food item is expired.
     * @param curDate The day that it currently is, as an int
     * @return True if expired, false if still good.
     */
    public boolean isExpired(int curDate) {
        return curDate >= expiryDate;
    }

    public FoodType getType() {
        return type;
    }

}
```