CoffeeMaker.java

```
   package edu.towson.cis.cosc603.project5.coffeemaker;
   /**
    * CoffeeMaker object
    * @version $Revision: 1.0 $
    */
   public class CoffeeMaker {
       /** Array of recipes in coffee maker */
       final private Recipe [] recipeArray;
       /** Number of recipes in coffee maker */
       private final int NUM_RECIPES = 4;
       /** Array describing if the array is full */
       final private boolean [] recipeFull;
       /** Inventory of the coffee maker */
       final private Inventory inventory;

     /**
      * Constructor for the coffee maker
      *
      */
       public CoffeeMaker() {
         recipeArray = new Recipe[NUM_RECIPES];
         recipeFull = new boolean[NUM_RECIPES];
3          for(int i = 0; i < NUM_RECIPES; i++) {
             recipeArray[i] = new Recipe();
             recipeFull[i] = false;
           }
           inventory = new Inventory();
       }

       /**
        * Returns true if a recipe is successfully added to the
        * coffee maker
        * @param r

        * @return boolean
        */
       public boolean addRecipe(Recipe r) {
       boolean canAddRecipe = true;

       canAddRecipe = checkExistenceOfRecipe( r );

       //Check for an empty recipe, add recipe to first empty spot
1      if(canAddRecipe) {
           int emptySpot = -1;
3          for(int i = 0; i < NUM_RECIPES; i++) {
1              if(!recipeFull[i]) {
                   emptySpot = i;
                   canAddRecipe = true;
               }
           }
1          if(emptySpot != -1) {
               recipeArray[emptySpot] = r;
               recipeFull[emptySpot] = true;
           }
           else {
               canAddRecipe = false;
           }
       }
1      return canAddRecipe;
     }

       /**
        *
        * Checks if the recipe already exists
```

```
              * @param r
              * @return boolean
              */
            private boolean checkExistenceOfRecipe(Recipe r) {

                    boolean canAddRecipe = true;

                    //Check if the recipe already exists
  3         for(int i = 0; i < NUM_RECIPES; i++) {
  1             if(r.equals(recipeArray[i])) {
                    canAddRecipe = false;
                }
            }
  1                 return canAddRecipe;
            }

            /**
             * Returns true if the recipe was deleted from the
             * coffee maker
             * @param r

             * @return boolean */
        public boolean deleteRecipe(Recipe r) {
            boolean canDeleteRecipe = false;
  1         if(r != null) {
  3                 for(int i = 0; i < NUM_RECIPES; i++) {
  1                     if(r.equals(recipeArray[i])) {
                            //Removed next line because purpose wasn't found
                            //recipeArray[i] = recipeArray[i];
                            canDeleteRecipe = true;
                        }
                    }
            }
  1         return canDeleteRecipe;
        }

        /**
         * Returns true if the recipe is successfully edited
         * @param oldRecipe
         * @param newRecipe

         * @return boolean */
        public boolean editRecipe(Recipe oldRecipe, Recipe newRecipe) {
            boolean canEditRecipe = false;
  3         for(int i = 0; i < NUM_RECIPES; i++) {
  1                 if(recipeArray[i].getName() != null) {
  1                     if(newRecipe.equals(recipeArray[i])) {
                            recipeArray[i] = new Recipe();
  1                         if(addRecipe(newRecipe)) {
                                    canEditRecipe = true;
                            }
                            else {
                                    canEditRecipe = false;
                            }
                        }
                    }
            }
  1         return canEditRecipe;
        }

        /**
         * Returns true if inventory was successfully added
         * @param amtCoffee
         * @param amtMilk
         * @param amtSugar
         * @param amtChocolate

         * @return boolean */
        public boolean addInventory(int amtCoffee, int amtMilk, int amtSugar, int amtChocolate) {
            boolean canAddInventory = true;
 8          if(amtCoffee < 0 || amtMilk < 0 || amtSugar < 0 || amtChocolate < 0) {
                canAddInventory = false;
            }
            else {
 2                  inventory.setCoffee(inventory.getCoffee() + amtCoffee);
 2                  inventory.setMilk(inventory.getMilk() + amtMilk);
 2                  inventory.setSugar(inventory.getSugar() + amtSugar);
 2                  inventory.setChocolate(inventory.getChocolate() + amtChocolate);
            }
 1          return canAddInventory;
        }

        /**
         * Returns the inventory of the coffee maker

         * @return Inventory */
        public Inventory checkInventory() {
 1          return inventory;
        }

        /**
         * Returns the change of a user's beverage purchase, or
         * the user's money if the beverage cannot be made
         * @param r
         * @param amtPaid

         * @return int */
        public int makeCoffee(Recipe r, int amtPaid) {
            boolean canMakeCoffee = true;
 2          if(amtPaid < r.getPrice()) {
                canMakeCoffee = false;
            }
 1          if(!inventory.enoughIngredients(r)) {
                canMakeCoffee = false;
            }
 1          if(canMakeCoffee) {
 2                  inventory.setCoffee(inventory.getCoffee() - r.getAmtCoffee());
 2                  inventory.setMilk(inventory.getMilk() - r.getAmtMilk());
 2                  inventory.setSugar(inventory.getSugar() - r.getAmtSugar());
 2                  inventory.setChocolate(inventory.getChocolate() - r.getAmtChocolate());
 2              return amtPaid - r.getPrice();
            }
            else {
 1              return amtPaid;
            }
        }

        /**
         * Returns an array of all the recipes

         * @return Recipe[] */
        public Recipe[] getRecipes() {
 1          return recipeArray;
        }

        /**
         * Returns the Recipe associated with the given name
         * @param name

         * @return Recipe */
            public Recipe getRecipeForName(String name) {
                    Recipe r = new Recipe();
 197 3          for(int i = 0; i < NUM_RECIPES; i++) {
 198 1              if(recipeArray[i].getName() != null) {
 199 1                  if((recipeArray[i].getName()).equals(name)) {
 200                        r = recipeArray[i];
 201                    }
 202                }
 203            }
 204 1          return r;
 205        }
 206    }
```

Mutations

23	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
43	1. negated conditional → KILLED
45	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
46	1. negated conditional → KILLED
51	1. negated conditional → KILLED
59	1. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
73	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
74	1. negated conditional → KILLED
78	1. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
89	1. negated conditional → KILLED
90	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
91	1. negated conditional → KILLED
98	1. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
109	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
110	1. negated conditional → KILLED
111	1. negated conditional → KILLED
113	1. negated conditional → KILLED
122	1. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
135	1. changed conditional boundary → KILLED 2. changed conditional boundary → KILLED 3. changed conditional boundary → KILLED 4. changed conditional boundary → KILLED 5. negated conditional → KILLED 6. negated conditional → KILLED 7. negated conditional → KILLED 8. negated conditional → KILLED
139	1. Replaced integer addition with subtraction → KILLED 2. removed call to edu/towson/cis/cosc603/project5/coffeemaker/Inventory::setCoffee → KILLED
140	1. Replaced integer addition with subtraction → KILLED 2. removed call to edu/towson/cis/cosc603/project5/coffeemaker/Inventory::setMilk → KILLED
141	1. Replaced integer addition with subtraction → KILLED 2. removed call to edu/towson/cis/cosc603/project5/coffeemaker/Inventory::setSugar → KILLED
142	1. Replaced integer addition with subtraction → KILLED 2. removed call to edu/towson/cis/cosc603/project5/coffeemaker/Inventory::setChocolate → KILLED
144	1. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
152	1. mutated return of Object value for edu/towson/cis/cosc603/project5/coffeemaker/CoffeeMaker::checkInventory to (if (x != null) null else throw new RuntimeException) → KILLED
164	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
167	1. negated conditional → KILLED
170	1. negated conditional → KILLED
171	1. Replaced integer subtraction with addition → KILLED 2. removed call to edu/towson/cis/cosc603/project5/coffeemaker/Inventory::setCoffee → KILLED
172	1. Replaced integer subtraction with addition → KILLED 2. removed call to edu/towson/cis/cosc603/project5/coffeemaker/Inventory::setMilk → KILLED
173	1. Replaced integer subtraction with addition → KILLED 2. removed call to edu/towson/cis/cosc603/project5/coffeemaker/Inventory::setSugar → KILLED

174	1. Replaced integer subtraction with addition → KILLED 2. removed call to edu/towson/cis/cosc603/project5/coffeemaker/Inventory::setChocolate → KILLED
175	1. Replaced integer subtraction with addition → KILLED 2. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
178	1. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
187	1. mutated return of Object value for edu/towson/cis/cosc603/project5/coffeemaker/CoffeeMaker::getRecipes to (if (x != null) null else throw new RuntimeException) → KILLED
197	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
198	1. negated conditional → KILLED
199	1. negated conditional → KILLED
204	1. mutated return of Object value for edu/towson/cis/cosc603/project5/coffeemaker/CoffeeMaker::getRecipeForName to (if (x != null) null else throw new RuntimeException) → KILLED

Active mutators

- INCREMENTS_MUTATOR
- VOID_METHOD_CALL_MUTATOR
- RETURN_VALS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- INVERT_NEGS_MUTATOR
- CONDITIONALS_BOUNDARY_MUTATOR

Tests examined

- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsGoodChocolate5(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEqualsRecipeSame(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtMilko(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadChocolate0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEditRecipe2(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetAmtMilk(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testDeleteRecipeNull(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddInventory1(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (14 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetCoffeeWithPos(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetChocolate(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (8 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetPrice0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testToStringRecipe(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsGood(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadSugar3(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetMilkWithNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtChocolateNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testPurchaseBeveragePriceNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testPurchaseBeverage3(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testPurchaseBeverageSugar0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (4 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetMilkWithPos(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtCoffee0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetCoffeeWithNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testCheckInventory(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddRecipeCopy(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddInventoryFailNegMilk(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (4 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtChocolatePos(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadChocolateNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testPurchaseBeverageGoodBuyWithChange(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetName(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (8 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testCheckInventorySetSugarNeg10(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetMilk(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtSugar0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtMilkNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadMilk3(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetSugarWith0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadCoffee0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddInventoryNegChocolate(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (10 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetMilkWith0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testCheckInventorySetCoffeeNeg10(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (6 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddInventoryPass0Coffee(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (5 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetChocolateWith0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetRecipes(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (3 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtMilkPos(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testCheckInventorySetChocolate0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetSugarWithNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddRecipeGood(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddInventoryFailNegCoffee(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (3 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddInventoryPass0Chocolate(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (3 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddRecipeAddTooMany(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (3 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadCoffee3(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (6 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetRecipeForName1(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (6 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadSugarNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetCoffeeWith0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (0 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetSugar(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtCoffeePos(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadMilkNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testPurchaseBeverageBadBuyNotEnoughMoney(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsGoodCoffee5(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddInventoryPass0Sugar(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadSugar5(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testPurchaseBeverage5(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadMilk0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEqualsRecipeThisNull(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (8 ms)

- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetPrice(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtCoffeeNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadCoffeeNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (3 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testPurchaseBeverageSugarTooLow(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testDeleteRecipeGood(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testPurchaseBeverageSugarNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (3 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtChocolate0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetSugarWithPos(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtSugarPos(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (4 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEditRecipe1(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetName(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetRecipeForName2(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (5 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadChocolate3(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testPurchaseBeverageGppdBuyExactChange(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (3 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadSugar0(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testEnoughIngredientsBadMilk5(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testToStringInventory(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testDeleteRecipeNotThere(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddInventoryNegSugar(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (4 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetAmtSugar(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testCheckInventorySetMilkNeg10(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (3 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetCoffee(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (3 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetAmtCoffee(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetPricePos(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (9 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testGetAmtChocolate(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetChocolateWithNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetAmtSugarNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (2 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetChocolateWithPos(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testSetPriceNeg(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testAddInventoryPass0Milk(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (4 ms)
- edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest.testPurchaseBeverage4(edu.towson.cis.cosc603.project5.coffeemaker.CoffeeMakerTest) (1 ms)

Report generated by PIT 1.1.5