Design Document: Recipe Adjustment Calculator

Description of the program

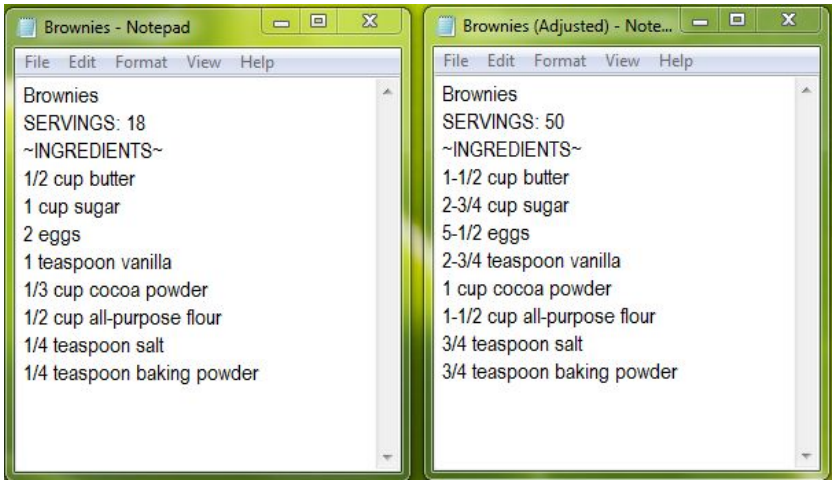
Recipe Adjustment Calculator reads recipes a recipe chosen by the user in from a file or creates a new one if they want to use their own recipe. Then it adjusts the recipe to any desired factor by printing the current recipe's serving size and asking the user how many servings they would like to serve. It rounds the amounts and units to simpler measurements that are helpful in the kitchen.

For example, if 2 ½ tbsp of sugar are required in a recipe that makes 6 servings but we want to serve 40, the program multiplies 2 ½ by 40/6 to get 50/3 tbsp which converts to 16 ⅔ tbsp. However, the program would print 1 cup after rounding because 16 tbsp = 1 cup.

The program currently has three predefined options from which user can choose:

- Chocolate Chip Cookies (serves 48)
- Brownies (serves 18)
- Create your own recipe

Here is an example of an input file (left) and the output file (right). As you can see, the original file has 18 servings while the adjusted recipe has 50. All the ingredient's amounts have been adjusted accordingly by multiplying all the original amounts by 50/18 → 25/9 (reduced).



You can also see the measurements have been rounded to easy to read numbers. For example, the last ingredient is ¼ teaspoon baking powder. In the new recipe, it should've displayed as 25/36 teaspoon which is around 0.694. However, when we're in the kitchen, those fractions will be no help as we don't always carry around a calculator. Therefore this program rounds it to ¾ teaspoon making it easier to measure.

Inputs to the program

- which recipe the user would like to see
    - if the user chooses one of the existing files, the file is prefilled with the recipe name, number of servings and ingredients listed with their amount as a fraction, unit (if applicable) and ingredient name
    - if the user chooses to use their own recipe the program asks for
        - the recipe name
        - number of servings their recipe has
        - number of ingredients
        - the numerator and denominator of the amount of their ingredient as an improper fraction
        - the unit of the ingredient (ex. teaspoon, tablespoon) if applicable
        - name of the ingredient
- how many new servings the user wants

Outputs to the program

- a file filled with
    - the recipe's title
    - new number of servings as requested by the user
    - adjusted amounts and units (if applicable and rounded to easy to read measurements used in the kitchen)
    - the ingredient name

Classes

Fraction class

Fields	
int numerator, denominator	The numerator and denominator of the fraction

Methods	
public **Fraction**(int n, int d)	Constructs a fraction with the numerator and denominator.

public Fraction **reduce**()	Reduces the fractions by dividing the numerator and denominator by their greatest common factor.
public double **toDecimal**()	Converts the fraction to a decimal.
public String **display**()	Displays the fraction in the form "numerator/denominator". Only returns the numerator if the denominator is equal to one. If the fraction is improper, it calls the MixedFraction class's display method.
public Fraction **multiply** (Fraction other)	Multiplies two fractions together and returns the product as a fraction.
public Fraction **multiply** (int constant)	Multiplies the fraction by a constant and returns the product as a fraction.
public Fraction **divide** (int constant)	Divides the fraction by a constant and returns the quotient as a fraction.
public static int **getGCD** (int a, int b)	Returns the GCD between two integers calculated recursively. Only used in the reduce method.
public static Fraction **decimalToFraction** (double d)	Converts a decimal into a fraction. This method is static because it doesn't use any of the fields and the user doesn't need to create a new Fraction object to use it.
private static boolean **isWholeNumber**(double d)	Returns true if the number in the parameter is a whole number. This method is private because it's only used by other methods in the class.
private MixedFraction **toMixedFraction** (int n, int d)	Converts an improper fraction into a mixed fraction.

MixedFraction class (extends Fraction)

Fields	
int whole	The whole number part of a mixed fraction.

Methods	
public **MixedFraction**(int w, int n, int d)	Constructs a mixed fraction with the whole number, numerator and denominator.
public double **toDecimal**()	Converts the mixed fraction into a decimal.
public String **display**()	Displays the mixed fraction in the form "whole number-numerator/denominator". Only returns the whole number if the numerator is equal to zero.
public Fraction **toImproper**()	Converts a mixed fraction into an improper fraction.

Main class

Fields	
ArrayList <Fraction> **amount**	An ArrayList of all the original recipe's amounts as a Fraction object.
ArrayList <Fraction> **adjustedAmount**	An ArrayList of all the adjusted amounts as a Fraction object.
ArrayList <String> **unit**	An ArrayList of all the unit names as a string.
ArrayList <String> **ingredient**	An ArrayList of all the ingredient names as a string.
String [] **possibleUnit**	An array of all the units this program recognizes and converts to a simpler measurement. I chose to only include teaspoon, tablespoon and cup because these 3 are the most commonly used units of measurement in cooking and can easily be converted to each other (ex. 1 cup = 48 tsp and 1 cup = 16 tbsp)

Methods	
public int **catchIntError**(String str)	Makes sure the user inputs a positive integer. Otherwise, keep printing **str**, the input question, and to input again.

public double **roundToQuarter**(double d)	Rounds a decimal to the nearest 0.25. For example, 0.79 would round to 0.75. However, if the nearest decimal is 0.0 return **d** instead.
public void **separateIngredientsList**(String list)	Takes **list** and separates it into three parts: the amount, the unit and the ingredient name by calling **addAmount** and **addIngredient**. Units are directly added to the ArrayList unit in this method.
public void **addAmount** (String a)	Converts **a** into a fraction object and adds it to the ArrayList amount.
public void **addIngredient** (String a [])	Adds the ingredient name into the ArrayList ingredient.
public void **newAmount** (Fraction cf)	Calculates the new adjusted amount by multiplying values in the ArrayList amount by **cf**. Adds the new amount to the ArrayList adjustedAmount.
public void **conversion** (ArrayList<Fraction> a, ArrayList<String> u)	Converts the amount and unit for a few ingredients into a simpler measurement. Rounds amounts larger than ½ using **roundToQuarter**. It rounds ≈ 1 cup to 48 tsp, ≈ 3 tsp to 1 tbsp and ≈ 16 tbsp to 1 cup. Replaces the the amount in **a** and unit in **u** with the conversions.
public String **chooseRecipe**() throws IOException	Asks the user which recipe they would like to see by entering a, b or c. If the user enters c, it proceeds to ask a series of questions so the program can create a new recipe file to read from. Adds the appropriate information to the new file based on the user's input. It returns the name of the recipe.
public static void **main**(String[] args) throws IOException	Reads the recipe file chosen and asks the user how many new serving they want their recipe to make. Prints the title, user's input number of servings and ingredients heading to the new file. Calls **separateIngredientsList**, **newAmount** and **conversion** to calculate the new adjusted amounts and units. It then prints the values in **adjustedAmount**, **unit** and **ingredient** to the new file.