package com.suarez;

import javax.swing.\*;

import java.util.\*;

/\*

Aarian Dhanani

2/18/19

This is ran when the user firsts opens the "tablet". Allows the user to choose their integer password

\*/

public class login {

//declared as public static so that it can be changed in settings

public static int *password*;

public static void password() {

int check = 0;

int chances = 3;

//Password registration

// System.out.println("Please enter your new password as an integer value");

String passwordJoption1= JOptionPane.*showInputDialog*("Please enter your new password as an integer value");

int temporaryJoption1 = Integer.*parseInt*(passwordJoption1);

// Scanner pass = new Scanner(System.in);

// password = pass.nextInt();

*password* = temporaryJoption1;

//login again to ensure they remember it

// System.out.println("Please log in again");

String passwordJoption2 = JOptionPane.*showInputDialog*("Please log in again");

int temporaryJoption2 = Integer.*parseInt*(passwordJoption2);

// Scanner guess = new Scanner(System.in);

// int guesses = pass.nextInt();

int guesses = temporaryJoption2;

while (*password* != guesses) {

//user has 4 chances to get the password correct or else it exits the entire program

while (check == 0)

{

if (chances != 0)

{

chances--;

System.*out*.println("Incorrect password. You have " + (chances + 1) + " chances remaining!");

}

else

{

System.*out*.println("Password entry failure. Your device will now reset");

System.*exit*(5);

}

String passwordJoption3 = JOptionPane.*showInputDialog*("Please enter: ");

int temporaryJoption3 = Integer.*parseInt*(passwordJoption3);

guesses = temporaryJoption3;

// guess = new Scanner(System.in);

// guesses = pass.nextInt();

if (*password* == guesses) //if password is correct

{

System.*out*.println("Welcome");

check++;

}

}

}

}

}

package com.suarez;

import java.awt.\*;

import java.io.File;

import java.util.Scanner;

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.event.ActionListener;

import java.util.concurrent.TimeUnit;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JTextArea;

/\*

Aarian Dhanani

2/18/19

Main area of the "tablet". This is where all of the classes are called and ran.

\*/

public class Main extends JFrame

{

public static void main(String[] args) {

// String test1= JOptionPane.showInputDialog("Please input mark for test 1: ");

// int int1 = Integer.parseInt(test1);

//Variable Setup:

int on = 1;

//1 time on open

System.*out*.println("Welcome to your Tablet");

login first = new login();

first.*password*();

//Master Menu Controller

while (on == 1) //continues looping the program

{

// System.out.println("Welcome to the main menu. Please enter which application you would like to open: \n[1]settings \n[2]calculator \n[3]contacts \n[4]app store \n[5]other apps");

// Scanner choose = new Scanner(System.in);

// int choice = choose.nextInt();

String menu5 = JOptionPane.*showInputDialog*("Welcome to the main menu. Please enter which application you would like to open: \n[1]settings \n[2]calculator \n[3]contacts \n[4]app store \n[5]other apps \n[6]shopping list");

int choice = Integer.*parseInt*(menu5);

switch(choice) //switch statement for all of the central classes

{

case 1:

*runSettings*();

break;

case 2:

*runcalculator*();

break;

case 3:

*runcontacts*();

break;

case 4:

*runappstore*();

break;

case 5:

//other apps work through system.out.print while all other parts work through JOptionPane

*runotherapps*();

break;

case 6:

*runshopping*();

break;

default: System.*out*.println("Invalid input!");

}

}

}

//Running programs in the switch statement for organization purposes

public static void runSettings()

{

settings first = new settings();

first.*settingsopen*();

}

public static void runcalculator()

{

calculator second = new calculator();

second.*calc*();

}

public static void runappstore()

{

appstore third = new appstore();

third.*appinstall*();

}

public static void runotherapps()

{

otherapps second = new otherapps();

second.*appsopen*();

}

public static void runcontacts()

{

contactsclass second = new contactsclass();

second.*maincontacts*();

}

public static void runshopping()

{

shoppingList trying = new shoppingList();

trying.fillWithList();

}

}

package com.suarez;

import javax.swing.\*;

import java.util.Scanner;

/\*

Aarian Dhanani

2/18/19

The settings class where preferences can be changed.

\*/

public class settings {

public static void settingsopen() {

//for easy exit later

int settingson = 1;

while (settingson == 1)

{

//allows user to choose which option they would like to go to

String menusettings = JOptionPane.*showInputDialog*("Welcome to settings. Please enter which option you would like to open: \n[1]delete apps \n[2]change password \n[3]reset \n[4]return to home page");

int choices = Integer.*parseInt*(menusettings);

switch (choices) {

//allows the user to set the values in enabledDisabled to 0 if it is currently 1

case 1:

JOptionPane.*showMessageDialog*(null, "Here is a list of your active 3rd party apps: ");

// System.out.println("Here is a list of your active 3rd party apps:");

enabledDisabled games = new enabledDisabled();

int game1 = games.*higherlower*;

int game2 = games.*rockpaperscissors*;

if (game1 == 1) {

JOptionPane.*showMessageDialog*(null, "Number Guessing Game\n");

// System.out.println("Number Guessing Game\n");

String menuopen = JOptionPane.*showInputDialog*("Would you like to delete Rock Paper Scissors?\n[1]yes\n[2]no");

int delete2 = Integer.*parseInt*(menuopen);

// System.out.println("Would you like to delete Number Guessing Game?\n[1]yes\n[2]no");

// Scanner deleteorno = new Scanner(System.in);

// int delete = deleteorno.nextInt();

switch (delete2) {

case 1:

System.*out*.println("Game deleted!");

games.*higherlower* = 0;

break;

case 2:

System.*out*.println("Canceled");

break;

default:

System.*out*.println("Invalid input!");

}

}

if (game2 == 1) {

JOptionPane.*showMessageDialog*(null, "Rock Paper Scissors\n");

// System.out.println("Rock Paper Scissors\n");

String menu50 = JOptionPane.*showInputDialog*("Would you like to delete Rock Paper Scissors?\n[1]yes\n[2]no");

int delete = Integer.*parseInt*(menu50);

// System.out.println("Would you like to delete Rock Paper Scissors?\n[1]yes\n[2]no");

// Scanner deleteorno = new Scanner(System.in);

// int delete = deleteorno.nextInt();

switch (delete) {

case 1:

JOptionPane.*showMessageDialog*(null, "Game deleted!");

// System.out.println("Game deleted!");

games.*rockpaperscissors* = 0;

break;

case 2:

JOptionPane.*showMessageDialog*(null, "Canceled");

// System.out.println("Canceled");

break;

default:

JOptionPane.*showMessageDialog*(null, "Invalid input!");

// System.out.println("Invalid input!");

}

}

if (game1 != 1 && game2 != 1) {

JOptionPane.*showMessageDialog*(null, "No other apps are enabled! Visit the app store to install.\n");

// System.out.println("No other apps are enabled! Visit the app store to install.\n");

}

break;

//allows user to change the value of password that was originally set

case 2:

login passwords = new login();

int passwordtest = passwords.*password*;

String passwordJoption2 = JOptionPane.*showInputDialog*("This area is password protected. Please enter your password.");

int temporaryJoption2 = Integer.*parseInt*(passwordJoption2);

int check = 0;

int chances = 3;

int guesses = temporaryJoption2;

while (passwordtest != guesses) {

//user has 4 chances to get the password correct or else it exits the entire program

while (check == 0)

{

if (chances != 0)

{

chances--;

System.*out*.println("Incorrect password. You have " + (chances + 1) + " chances remaining!");

}

else

{

System.*out*.println("Password entry failure. Your device will now reset");

System.*exit*(5);

}

String passwordJoption3 = JOptionPane.*showInputDialog*("Please enter: ");

int temporaryJoption3 = Integer.*parseInt*(passwordJoption3);

guesses = temporaryJoption3;

if (passwordtest == guesses) //if password is correct

{

System.*out*.println("Welcome");

check++;

}

}

}

login first = new login();

first.*password*();

System.*out*.println("Password changed!");

break;

//allows user to reset their device (exits the entire program)

case 3:

login passwords1 = new login();

int passwordtest1 = passwords1.*password*;

String passwordJoption21 = JOptionPane.*showInputDialog*("This area is password protected. Please enter your password.");

int temporaryJoption21 = Integer.*parseInt*(passwordJoption21);

int check1 = 0;

int chances1 = 3;

int guesses1 = temporaryJoption21;

while (passwordtest1 != guesses1) {

//user has 4 chances to get the password correct or else it exits the entire program

while (check1 == 0)

{

if (chances1 != 0)

{

chances1--;

System.*out*.println("Incorrect password. You have " + (chances1 + 1) + " chances remaining!");

}

else

{

System.*out*.println("Password entry failure. Your device will now reset");

System.*exit*(5);

}

String passwordJoption31 = JOptionPane.*showInputDialog*("Please enter: ");

int temporaryJoption31 = Integer.*parseInt*(passwordJoption31);

guesses = temporaryJoption31;

if (passwordtest1 == guesses) //if password is correct

{

System.*out*.println("System reset starting");

check1++;

System.*exit*(10);

}

}

}

break;

//returns the user to the main menu

case 4:

System.*out*.println("You will now return to the main menu.");

settingson = 0;

break;

default: System.*out*.println("Invalid input!");

}

}

}

}

package com.suarez;

import javax.swing.\*;

import java.util.\*;

//adapted from https://stackoverflow.com/questions/14675815/basic-calculator-in-java

/\*

Aarian Dhanani

2/18/19

Run a simple calculator for the tablet

\*/

public class calculator {

public static void calc()

{

int opencalc = 1; //Sets the value to open so that it can be easily closed later on

int integer1;

int integer2;

System.*out*.println("Welcome to text based integer calculator");

while (opencalc == 1) //sets up so that once it is closed later, it can break from the while loop

{

Scanner numbers = new Scanner(System.*in*);

//gets the integers

String opencalculator= JOptionPane.*showInputDialog*("Please enter the 1st number: ");

integer1 = Integer.*parseInt*(opencalculator);

String opencalculator2= JOptionPane.*showInputDialog*("Please enter the 2nd number: ");

integer2 = Integer.*parseInt*(opencalculator2);

//old code for text based:

// System.out.println("Please enter the 1st integer value");

// integer1 = numbers.nextInt();

// System.out.println("Please enter the 2nd integer value");

// integer2 = numbers.nextInt();

//allows the user to perform an operation

String calculator = JOptionPane.*showInputDialog*("Which operation would you like to perform? \n[1]addition\n[2]subtraction\n[3]multiplication\n[4]division");

int choice = Integer.*parseInt*(calculator);

switch (choice){

case 1:

JOptionPane.*showMessageDialog*(null, integer1 + integer2);

// System.out.println(integer1 + integer2);

break;

case 2:

JOptionPane.*showMessageDialog*(null, integer1 - integer2);

// System.out.println(integer1 - integer2);

break;

case 3:

JOptionPane.*showMessageDialog*(null, integer1 \* integer2);

// System.out.println(integer1 \* integer2);

break;

case 4:

if (integer2==0)

{

JOptionPane.*showMessageDialog*(null, "Invalid operation.");

}

else

{

JOptionPane.*showMessageDialog*(null, integer1 / integer2);

}

// System.out.println(integer1 / integer2);

break;

default:

JOptionPane.*showMessageDialog*(null, "Invalid operation.");

// System.out.println("Invalid operation!");

}

//Asks to see if the loop should be ran again

String redo = JOptionPane.*showInputDialog*("Would you like to perform another operation?\n[1]Yes\n[2]No");

choice = Integer.*parseInt*(redo);

switch (choice){

case 1:

break;

case 2:

JOptionPane.*showMessageDialog*(null, "You will now return to the main menu.");

// System.out.println("You will now return to the main menu.");

opencalc = 0; //exits the loop and class

break;

default:

JOptionPane.*showMessageDialog*(null, "Invalid input!");

// System.out.println("Invalid input!");

}

}

}

}

package com.suarez;

/\*

Aarian Dhanani

2/18/19

Install 3rd party apps for the tablet

\*/

import javax.swing.\*;

import java.util.Scanner;

public class appstore {

public static void appinstall() {

//Calling the values of the 2 games

int openedentry = 0;

enabledDisabled games = new enabledDisabled();

int game1 = games.*higherlower*;

int game2 = games.*rockpaperscissors*;

//Always opens this. Shows which apps are not installed

JOptionPane.*showMessageDialog*(null, "Here is a list of 3rd party apps ready to install:");

// System.out.println("Here is a list of 3rd party apps ready to install:");

if (game1 == 0) {

JOptionPane.*showMessageDialog*(null, "Number Guessing Game");

// System.out.println("Number Guessing Game");

openedentry = 1;

}

if (game2 == 0) {

JOptionPane.*showMessageDialog*(null, "Rock Paper Scissors");

// System.out.println("Rock Paper Scissors");

openedentry = 1;

}

if (game1 == 1 && game2 == 1) {

JOptionPane.*showMessageDialog*(null, "All 3rd party apps are installed. You will now return to the main menu");

//this is for if a game has not installed

// System.out.println("All 3rd party apps are installed. You will now return to the main menu");

}

//This part only shows if there is a game that has not been installed

if (openedentry == 1)

{

String yesorno= JOptionPane.*showInputDialog*("Would you like to install a game?\n[1]yes\n[2]no");

int install = Integer.*parseInt*(yesorno);

// System.out.println("Would you like to install a game?\n[1]yes\n[2]no"); //Checks to make sure the user wants to install a game

// Scanner installorno = new Scanner(System.in);

// int install = installorno.nextInt();

switch (install) {

case 1:

if (game1 == 0 && game2 ==0)

{

//Allows you to choose from both

String yesorno1= JOptionPane.*showInputDialog*("Which would you like to install?\n[1]Number Guessing Game\n[2]Rock Paper Scissors\n[3]Cancel");

int install1 = Integer.*parseInt*(yesorno1);

if (install1 == 1)

{

games.*higherlower* = 1;

JOptionPane.*showMessageDialog*(null, "Installed. You will now return to the main menu");

}

else if (install1 == 2)

{

games.*rockpaperscissors* = 1;

JOptionPane.*showMessageDialog*(null, "Installed. You will now return to the main menu");

}

openedentry--;

}

else if (game1 == 0)

{

// System.out.println("Would you like to install Number Guessing Game?\n[1]Yes\n[2]No");

// int install2 = installorno.nextInt();

String yesorno2= JOptionPane.*showInputDialog*("Would you like to install a game?\n[1]yes\n[2]no");

int install2 = Integer.*parseInt*(yesorno2);

if (install2 == 1)

{

games.*higherlower* = 1;

JOptionPane.*showMessageDialog*(null, "Installed. You will now return to the main menu");

}

openedentry--;

}

else if (game2 == 0)

{

String yesorno3= JOptionPane.*showInputDialog*("Would you like to install Rock Paper Scissors?\n[1]Yes\n[2]No");

int install3 = Integer.*parseInt*(yesorno3);

// System.out.println("Would you like to install Rock Paper Scissors?\n[1]Yes\n[2]No");

// int install3 = installorno.nextInt();

if (install3 == 1)

{

games.*rockpaperscissors* = 1;

JOptionPane.*showMessageDialog*(null, "Installed. You will now return to the main menu");

}

openedentry--;

}

break;

case 2:

//If the user inputs canceled it breaks from the class and returns to the main menu

JOptionPane.*showMessageDialog*(null, "Canceled. You will now return to the main menu.");

// System.out.println("Canceled. You will now return to the main menu.");

break;

default:

JOptionPane.*showMessageDialog*(null, "Invalid input! You will now return to the main menu.");

// System.out.println("Invalid input! You will now return to the main menu.");

}

}

}

}

package com.suarez;

import javax.swing.\*;

import java.util.Scanner;

/\*

Aarian Dhanani

2/18/19

Where the computer checks if the games are enabled and if playable, then the user can choose to play them.

\*/

public class otherapps {

public static void appsopen()

{

//calling values from enabledDisabled to check later if games are playable

enabledDisabled games = new enabledDisabled();

int game1 = games.*higherlower*;

int game2 = games.*rockpaperscissors*;

Scanner startgameorno = new Scanner(System.*in*);

//checks which games are playable and allows user to play the enabled ones

if (game1 == 1 && game2 ==1)

{

// System.out.println("Number Guessing Game AND Rock Paper Scissors are enabled. Would you like to play one of them?\n[1]yes\n[2]no");

// int open = startgameorno.nextInt();

String opencheck= JOptionPane.*showInputDialog*("Number Guessing Game AND Rock Paper Scissors are enabled. Would you like to play one of them?\n[1]yes\n[2]no");

int open = Integer.*parseInt*(opencheck);

if (open == 1)

{

String opencheck2= JOptionPane.*showInputDialog*("Which would you like to play?\n[1]Number Guessing Game\n[2]Rock Paper Scissors");

int open2 = Integer.*parseInt*(opencheck2);

// System.out.println("Which would you like to play?\n[1]Number Guessing Game\n[2]Rock Paper Scissors");

// int open2 = startgameorno.nextInt();

switch (open2){

case 1:

// System.out.println("Running Number Guessing Game.");

JOptionPane.*showMessageDialog*(null, "Running Number Guessing Game");

*runhigherlower*();

break;

case 2:

*runrockpaperscissors*();

break;

default:

JOptionPane.*showMessageDialog*(null, "Invalid input!");

}

}

else

{

JOptionPane.*showMessageDialog*(null, "Canceled");

// System.out.println("Canceled");

}

}

//only if guessing game is enabled but not rock paper scissors

else if (game1 == 1)

{

String opencheck3= JOptionPane.*showInputDialog*("Number Guessing Game is enabled. Would you like to play?\n[1]yes\n[2]no");

int open = Integer.*parseInt*(opencheck3);

// System.out.println("Number Guessing Game is enabled. Would you like to play?\n[1]yes\n[2]no");

// int open = startgameorno.nextInt();

if (open == 1){

*runhigherlower*();

}

else

{

JOptionPane.*showMessageDialog*(null, "Canceled");

}

}

//only if guessing game is not enabled but rock paper scissors is

else if (game2 == 1)

{

String opencheck4= JOptionPane.*showInputDialog*("Rock Paper Scissors is enabled. Would you like to play?\n[1]yes\n[2]no");

int open = Integer.*parseInt*(opencheck4);

// System.out.println("Rock Paper Scissors is enabled. Would you like to play?\n[1]yes\n[2]no");

// int open = startgameorno.nextInt();

if (open == 1){

*runrockpaperscissors*();

}

else

{

JOptionPane.*showMessageDialog*(null, "Canceled");

}

}

//if both are disabled

else if (game1 == 0 && game2 == 0)

{

JOptionPane.*showMessageDialog*(null, "No 3rd party apps have been installed. Please visit the app store to do so.");

// System.out.println("No 3rd party apps have been installed. Please visit the app store to do so.");

}

}

//area where the classes are run

public static void runhigherlower()

{

higherlower second = new higherlower();

second.*mainhigherlower*();

}

public static void runrockpaperscissors()

{

rockpaperscissors first = new rockpaperscissors();

first.*runrockpaperscissors*();

}

}

package com.suarez;

import java.util.Random;

import java.util.Scanner;

/\*

Aarian Dhanani

2/18/19

Where the rock paper scissors game can be played through other apps.

\*/

public class rockpaperscissors {

public static void runrockpaperscissors(){

//set up for easy exit out of class later

int runpaper = 1;

while(runpaper == 1)

{

//the computer generates a random number which symbolizes rock, paper, or scissors

Random rand = new Random();

int compchoice = rand.nextInt(3)+1;

// System.out.println(compchoice); Testing purposes only

//the user inputs a number which symbolizes rock, paper, or scissors

System.*out*.println("Choose: \n[1]rock\n[2]paper\n[3]scissors");

Scanner scan = new Scanner(System.*in*);

int playerchoice = scan.nextInt();

//depending on the user's choice and the computer's choice, it prints a statement

switch (playerchoice){

case 1:

if (compchoice == 1)

{

System.*out*.println("It is a tie!");

}

if (compchoice == 2)

{

System.*out*.println("You lose!");

}

if (compchoice ==3)

{

System.*out*.println("You win!");

}

break;

case 2:

if (compchoice == 2)

{

System.*out*.println("It is a tie!");

}

if (compchoice == 3)

{

System.*out*.println("You lose!");

}

if (compchoice ==1)

{

System.*out*.println("You win!");

}

break;

case 3:

if (compchoice == 3)

{

System.*out*.println("It is a tie!");

}

if (compchoice == 1)

{

System.*out*.println("You lose!");

}

if (compchoice ==2)

{

System.*out*.println("You win!");

}

break;

default:

System.*out*.println("Invalid input!");

}

//checks if the user would like to play again

System.*out*.println("Would you like to play again?\n[1]yes\n[2]no");

int playagain = scan.nextInt();

switch (playagain){

case 1:

break;

case 2:

System.*out*.println("You will now return to the main menu");

runpaper--;

break;

default:

System.*out*.println("Invalid input!");

}

}

}

}

package com.suarez;

/\*

Aarian Dhanani

2/18/19

Mini game where the user has to guess a random number

\*/

import java.util.Random;

import java.util.Scanner;

public class higherlower {

public static void mainhigherlower()

{

*write*();

}

//computer generates a random number

public static void write(){

Random rand = new Random();

int number = rand.nextInt(10)+1;

*check*(number); //goes into the main part of the code

}

//main part of code where user interacts

public static void check(int num1) {

//sets number of tries

System.*out*.println("How many tries would you like to play with?");

int guesses = 0;

Scanner guess = new Scanner(System.*in*);

guesses = guess.nextInt();

int scans = 0;

System.*out*.println("Please guess a number between 1 and 10");

while (scans != num1 && guesses > 0) //user continues guessing until they are out of tries

{

System.*out*.println("Give it a guess!");

Scanner scan = new Scanner(System.*in*);

scans = scan.nextInt();

if (scans != num1) {

if (Math.*abs*(num1 - scans) > 3) //gives user a hint on how far they are from the correct number

{

System.*out*.println("You were off by a mile!");

guesses = guesses -1;

}

else{

System.*out*.println("You were so close!");

guesses = guesses -1;

}

}

}

if (guesses > 0){

System.*out*.println("You WIN, Congrats! You will now return to the main menu.");

}

else{

System.*out*.println("Sorry, you are out of lives and lose! You will now return to the main menu");

}

}

}

package com.suarez;

import javax.swing.\*;

import java.util.ArrayList;

import java.util.Scanner;

import java.lang.String;

import java.io.\*;

import java.io.FileNotFoundException;

/\*

Aarian Dhanani

2/18/19

This is the main for the contacts. This is where the user interacts and can add contacts as well as view contacts.

\*/

//adapted from https://stackoverflow.com/questions/21664677/creating-a-contact-list-with-java-and-object-oriented

public class contactsclass

{

public static ArrayList<Contact> *contacts* = new ArrayList<Contact>();

public static void maincontacts() {

int contactson = 1;

while (contactson == 1)

{

int adding = 0; //to add a new contact to the ArrayList

//main area

String opencontacts= JOptionPane.*showInputDialog*("What would you like to do?\n[1]View Saved Contacts \n[2]Add Contacts\n[3]Save all new contacts\n[4]Return to the Main Menu");

int contactsoption = Integer.*parseInt*(opencontacts);

switch (contactsoption){

case 1:

//view contacts: in this new version of the tablet, you can actually look at contacts that have been exported!

try

{

Scanner inputContacts = new Scanner(new File("/Users/aariandhanani/IdeaProjects/Java2finalproject/contactsheet.txt"));

// ArrayList<String> contactslist2 = new ArrayList<String>();

// while (inputContacts.hasNext())

// {

// contactslist2.add(inputContacts.next());

// }

// inputContacts.close();

String showContacts = "";

String temporary = "";

String space = "";

while ( inputContacts.hasNext())

{

temporary = inputContacts.nextLine();

if (temporary.startsWith("Email Address:"))

{

space = "\n";

}

showContacts = showContacts + temporary + "\n" + space;

temporary = "";

space = "";

}

inputContacts.close();

JOptionPane.*showMessageDialog*(null, "Here are your registered contacts: \n" + showContacts);

}

catch (IOException ioe)

{

ioe.printStackTrace( );

}

break;

case 2:

//input contacts

String yesorno1= JOptionPane.*showInputDialog*("Would you like to enter a new contact?\n[1]yes\n[2]no");

int yesorno = Integer.*parseInt*(yesorno1);

switch (yesorno) {

case 1:

adding = 1;

break;

case 2:

JOptionPane.*showMessageDialog*(null, "You will now return to the menu!");

break;

default:

JOptionPane.*showMessageDialog*(null, "Invalid input!");

}

while (adding == 1) //if said yes to add a new contact

{

//Takes in name and email

String passwordJoptionname= JOptionPane.*showInputDialog*("Please enter the first name: ");

String name1 = passwordJoptionname;

String passwordJoptionemail= JOptionPane.*showInputDialog*("Please enter the email address: ");

String email1 = passwordJoptionemail;

Contact contact;

contact = new completeContact(name1, email1);

*contacts*.add(contact);

// System.out.println("New contact added: ");

// contact.displayContact();

String display = "New contact added: " + contact.displayContact2();

JOptionPane.*showMessageDialog*(null, display);

String yesorno2= JOptionPane.*showInputDialog*("Would you like to enter another contact?\n[1]yes\n[2]no");

int yesornocheck = Integer.*parseInt*(yesorno2);

switch (yesornocheck)

{

case 1:

break;

case 2:

JOptionPane.*showMessageDialog*(null, "You will now return to the contacts menu.");

adding = 0;

break;

default:

JOptionPane.*showMessageDialog*(null, "Invalid input!");

}

}

break;

case 3:

*exportcontact*();

break;

case 4:

JOptionPane.*showMessageDialog*(null, "You will now return to the main menu!");

contactson = 0;

break;

default:

JOptionPane.*showMessageDialog*(null, "Invalid input!");

}

}

}

public static void exportcontact() {

try

{

File contactsheet = new File("contactsheet.txt");

FileWriter contactfile = new FileWriter(contactsheet, true);

for ( Contact x: *contacts*){

contactfile.write(x.contactlistsheet());

}

contactfile.close();

JOptionPane.*showMessageDialog*(null, "Contacts Exported");

*contacts*.clear(); //clears the arraylist so that the same contacts cannot be exported again on accident

}

catch (IOException ioe)

{

System.*out*.println("There are no contacts!");

}

}

}

package com.suarez;

/\*

Aarian Dhanani

2/18/19

Sets up string values for name and email for the contacts so that it can easily be called in the main contacts class

\*/

public abstract class Contact

{

private String name;

private String email;

public Contact()

{

this.name = "NoOne";

this.email = "NoOne@noOne.com";

}

public Contact(String name, String email)

{

this.name = name;

this.email = email;

}

//sets the name

public void setName(String input)

{

this.name = input;

}

public String getName()

{

return name;

}

public void setemail(String input)

{

this.email = input;

}

public String getemail()

{

return email;

}

void displayContact()

{

System.*out*.println("\nName: " + name + "\nEmail Address: " + email);

}

public String displayContact2()

{

String words = "\nName: " + name + "\nEmail Address: " + email;

return words;

}

public String contactlistsheet()

{

return ("\nName: " + name + "\nEmail Address: " + email);

}

}

package com.suarez;

/\*

Aarian Dhanani

2/18/19

Set up and get all the String values for the contacts, just for simplicity and easy access

\*/

public class completeContact extends Contact {

public completeContact(String name, String email)

{

super(name, email);

}

}

package com.suarez;

import javax.swing.\*;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Scanner;

public class shoppingList

{

//object class for the shopping list

public static String *shoppingwords*;

private String item;

private int price;

public static ArrayList<shoppingList> *shopList*;

int again = 1;

public void initialize( )

{

*shopList* = new ArrayList<shoppingList>( );

}

public shoppingList()

{

item = "unkown";

}

public shoppingList(String item1) {

item = item1;

}

public shoppingList(String item1, int price) {

item = item1;

SetPrice(price);

}

public shoppingList SetPrice(int price)

{

if (price < 0)

{

this.price = 0;

}

else

{

this.price = price;

}

return this;

}

public String Getitem()

{

return item;

}

public int Getprice()

{

return price;

}

public String toString(String item, int price)

{

*shoppingwords* = "Item: " + item + "\nPrice: " + price;

return *shoppingwords*;

}

//due to how the tablet functions, it was much easier to include what would be the "client" after the object class

public void fillWithList( )

{

initialize();

again = 1;

int continueIt = 1;

while (again == 1)

{

String yesorno1= JOptionPane.*showInputDialog*("What would you like to do?\n[1]add an item \n[2]view list\n[3]save items\n[4]return to menu");

int yesorno = Integer.*parseInt*(yesorno1);

switch (yesorno)

{

case 1:

while(continueIt == 1)

{

String JoptionItem = JOptionPane.*showInputDialog*("Please enter the item name: ");

String name1 = JoptionItem;

String number= JOptionPane.*showInputDialog*("Please enter the price (to the nearest whole number)");

int num = Integer.*parseInt*(number);

// shoppingList item1 = new shoppingList( "" + name1, 0 + num);

// shopList.add(item1);

// JOptionPane.showMessageDialog(null, "Item saved!");

shoppingList shop;

shop = new shoppingList(name1, num);

*shopList*.add(shop);

// System.out.println("New contact added: ");

// contact.displayContact();

String display = "New item added: " + shop.shoppingListSheetWords();

JOptionPane.*showMessageDialog*(null, display);

String continue2 = JOptionPane.*showInputDialog*("Would you like to add another item?\n[1]yes\n[2]no ");

int yesorno10 = Integer.*parseInt*(continue2);

switch (yesorno10)

{

case 1:

break;

case 2:

continueIt = 0;

break;

default:

JOptionPane.*showMessageDialog*(null, "Invalid input!");

}

}

break;

case 2:

try

{

Scanner inputContacts = new Scanner(new File("/Users/aariandhanani/IdeaProjects/Java2finalproject/src/shoppinglist"));

String showContacts = "";

String temporary = "";

String space = "";

while ( inputContacts.hasNext())

{

temporary = inputContacts.nextLine();

if (temporary.startsWith("Email Address:"))

{

space = "\n";

}

showContacts = showContacts + temporary + "\n" + space;

temporary = "";

space = "";

}

inputContacts.close();

JOptionPane.*showMessageDialog*(null, "Here is your shopping list: \n" + showContacts);

}

catch (IOException ioe)

{

ioe.printStackTrace( );

}

break;

case 3:

*exportList*();

break;

case 4:

JOptionPane.*showMessageDialog*(null, "You will now return to the main menu!");

again = 0;

break;

default:

JOptionPane.*showMessageDialog*(null, "Invalid input!");

}

}

}

public static void exportList()

{

try

{

File shoppingSheet = new File("/Users/aariandhanani/IdeaProjects/Java2finalproject/src/shoppinglist");

FileWriter shoppingfile = new FileWriter(shoppingSheet, true);

for ( shoppingList x: *shopList*){

shoppingfile.write(x.shoppingListSheetWords());

}

shoppingfile.close();

JOptionPane.*showMessageDialog*(null, "Items added!");

*shopList*.clear(); //clears the arraylist so that the same items cannot be exported again on accident

}

catch (IOException ioe)

{

System.*out*.println("There are no items!");

}

}

public String shoppingListSheetWords()

{

return ("\nItem: " + item + "\nPrice: " + price);

}

}

package com.suarez;

/\*

Aarian Dhanani

2/18/19

This is where 3rd party apps are set to enabled/disabled. They are called in other classes

\*/

public class enabledDisabled {

public static int *higherlower* = 0;

public static int *rockpaperscissors* = 0;

}