**A picture containing qr code

Description automatically generated**

**Gadget Shop program with GUI Report**

**Andrius Uzkuraitis**

**CS4001: Programming**

Deadline: 06/05/2022

**Module Leader:** Sandra Fernando **Student ID:** **20028126**

**PLAGIARISM**

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed below. Please sign below to say that you have read and understand these extracts:

**Signature:**  **Date**:12/04/2022

Extracts from University *Regulations on*

Cheating, Plagiarism and Collusion

Section 2.3:

"The following broad types of offence can be identified and are provided as indicative examples ...

1. **Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.**
2. **Falsifying data in experimental results.**
3. Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.
4. **Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.**
5. Collusion to present joint work as the work solely of one individual.
6. Plagiarism, where the work or ideas of another are presented as the candidate's own.
7. Other conduct calculated to secure an advantage on assessment.
8. Assisting in any of the above.

Some notes on what this means for students:

Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.

Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Stating an author’s argument and attributing it, perhaps by quoting, achieve quoting ideas

Contents

[Class Diagram 1](#_Toc101137973)

[Button Handling and Methods Description 2](#_Toc101137974)

[Testing 6](#_Toc101137975)

[Error detection and correction 18](#_Toc101137976)

[Conclusion 20](#_Toc101137977)

[APPENDIX: PROJECT CODE 21](#_Toc101137978)

# Class Diagram

Diagram

Description automatically generated

# Button Handling and Methods Description

**1.Add Mobile**

**-**public void addMobile():

This method is used to create a new instance of a Mobile class using previously defined get methods that retrieve users input from the text fields. This method also adds the new instance of Mobile to the item ArrayList.

If value returned by getModel() , getPrice(), getWeight(), getSize(), getCredit() doesn’t meet the conditions{

Don’t add new Mobile to Item ArrayList

Display a pop-up window with error message that new Mobile is not added

}

Else {add new Mobile

Display a pop-up message that new Mobile is added

}

**2.Add Mp3**

**-**public void AddMp3():

This method is used to create a new instance of a Mp3 class using previously defined get methods that retrieve users input from the text fields. This method also adds the new instance of Mp3 to the item ArrayList.

If value returned by getModel(), getPrice(), getWeight(), getSize(), getMemory() doesn’t meet the conditions{

Don’t add new Mp3 to Item ArrayList

Display a pop-up window with error message that new Mp3 is not added

}

Else {add new Mp3

Display a pop-up message that new Mp3 is added

}

**3.Make A Call**

**-**public void makeACall():

This method is used to make a call using previously defined methods retrieve user input from text fields.

If getPhoneNumber(), getDuration(), getDisplayNumber() doesn’t meet the conditions{

Call is cancelled

Display a pop-up window with error message Call is cancelled

}

Else{

Try{

If{ getDisplayNumber () meets conditions Gadget in the ArrayList {

Call is made and display message with phone number and duration appears

}

Else{ if the display number is not a Mobile

Display a pop-up window with message that this display number is not Mobile

}

Catch IndexOutOfBoundsException exception{

If number doesn’t exist as a Mobile or Mp3 display pop-up message that gadget with this number doesn’t exist

}

**4.Dowload Music**

-public void DowloadMusic()

This method is used to download music using previously defined methods retrieve user input from text fields.

If getMusicfile(), getDisplayNumber() doesn’t meet the conditions {

Music download is cancelled

Display a pop-up window with error message Music download is cancelled

}

Else{

Try{

If{ getDisplayNumber () meets conditions with Gadget in the ArrayList {

Music is downloaded and display message with remaining memory and used memory appears.

}

Else{ if the display number is not a Mp3

Display a pop-up window with message that this display number is not Mp3

}

Catch IndexOutOfBoundsException exception{

If number doesn’t exist as a Mobile or Mp3 display pop-up message that gadget with this number doesn’t exist

}

**5.Display Number**

-public int getDisplayNumber():

Try{

If getDisplayNumber(), doesn’t meet the condition {

Number is not valid

Display pop-up message that number is not valid please check your entry

}

}

Catch NumberFormatException exception )if condition doesn’t meet {

Display pop-up message that display field is empty or format is wrong.

}

Else return get display number

**6.Display All**

-Public void DisplayAll()

this method is used to display all the cars currently stored in my ArrayList. It will first clear the terminal window, then perform a for loop.

ClearTerminal ();

For every gadget in ArrayList item

{

Print the index of this Gadget + 0 followed by a “:” ; (this gives us output of the gadget number in display)

Call the getdisplay() method from the gadget class;

Print an empty line;(for aesthetic reasons)

# Testing

**Test 1**: Adding a mobile to the array list

Pop-up display that new phone is added

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

If a field is left empty or a format of text is wrong pop-up message displayed followed by error message that new Mobile is not added

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

**Test 2**: Adding an MP3 player to the array list

Pop-up display that new Mp3 is added

Graphical user interface, application, Word

Description automatically generated

Graphical user interface

Description automatically generated

If a field is left empty or a format of text is wrong pop-up message displayed followed by error message that new Mp3 is not added

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

**Test 3**: Displaying the details of all of the gadgets in the array list

Graphical user interface

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

**Test 4**: Making a call

If Phone Number format and length is correct and there is enough credits and display number is right the display message appears

Graphical user interface, table

Description automatically generated with medium confidence

If there is not enough credit message displayed

Graphical user interface, application

Description automatically generated

If duration and phone number field is empty or format is wrong pop up message displayed followed by error message that call is cancelled

Graphical user interface, application

Description automatically generated

Graphical user interface

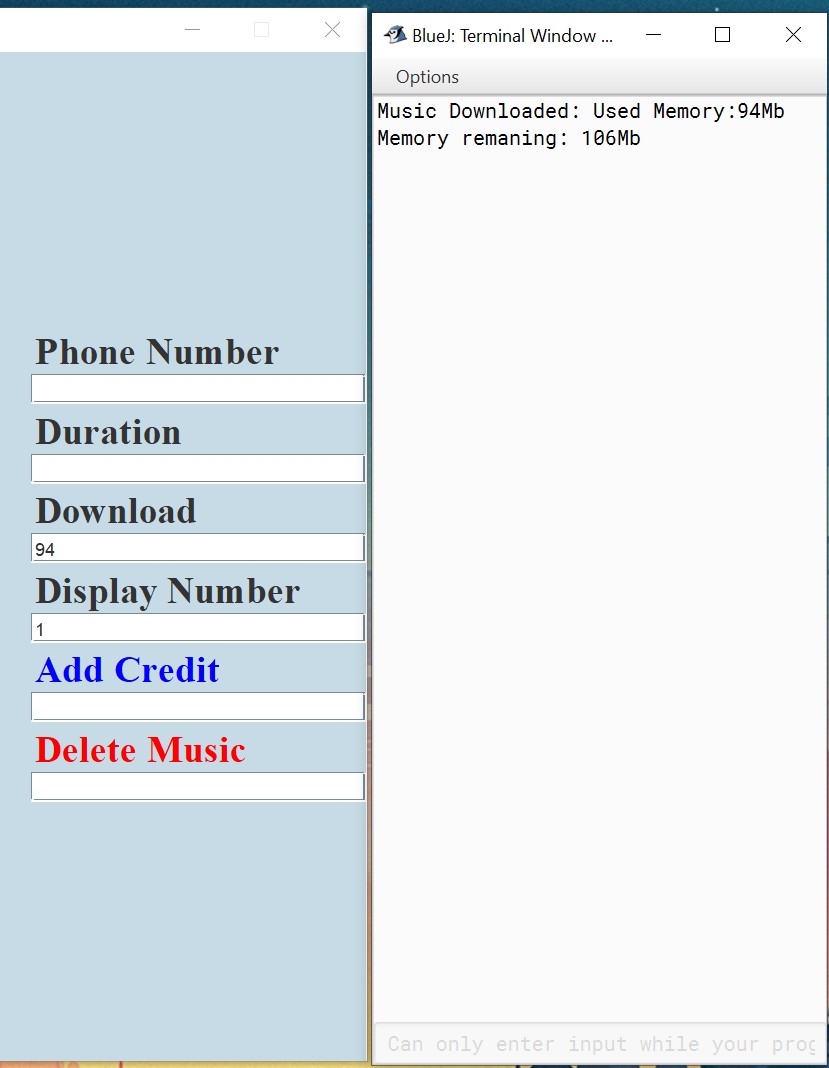
Description automatically generated

Graphical user interface, application, Word

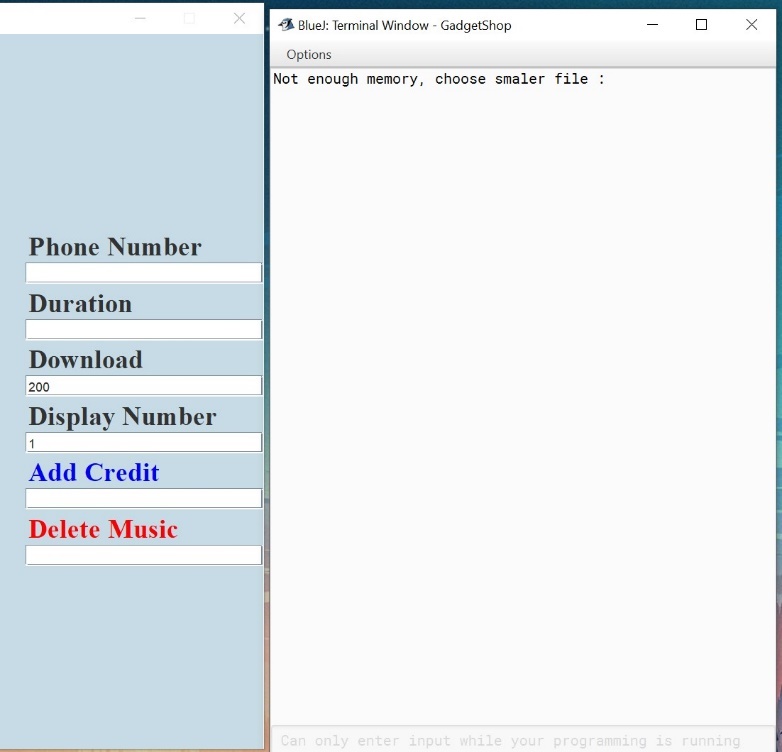
Description automatically generated

**Test 5**: Downloading music

If there is enough memory and the display number is correct



If there is not enough memory



If download field is empty or the format is wrong pop up message displayed followed by error message that download is cancelled

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

**Test 6**: Test that the program can be compiled and run using the command prompt, including a screenshot similar to Figure 1 in the command prompt learning aid.

Graphical user interface

Description automatically generated

**Test 7**: Test that appropriate dialog boxes appear when unsuitable values are entered for the display number.

(Include a screenshot of the dialog box, together with a corresponding screenshot of the GUI, showing the values that were entered.)

When making a call or and using Mp3 display number pop up message displayed

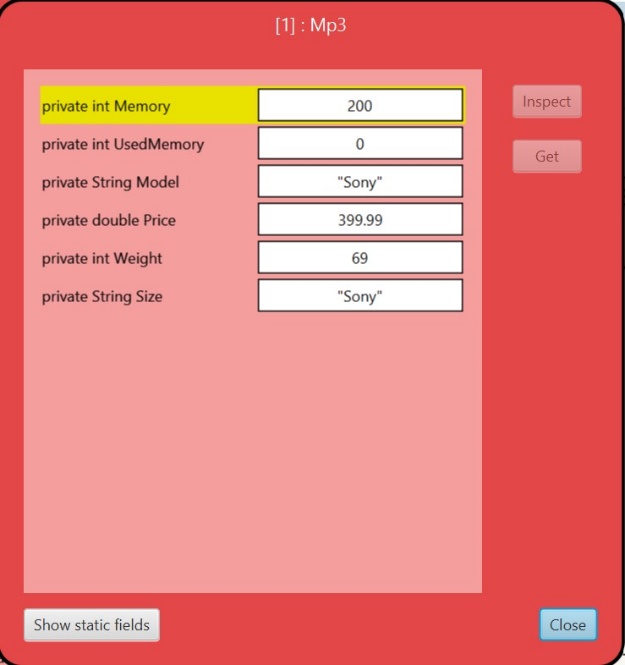
Graphical user interface, application

Description automatically generated

When downloading music using Mobile display number pop up message displayed

Graphical user interface, application, Word

Description automatically generated



When there is no gadget in ArrayList with the number provided. Pop-up message displayed.

Graphical user interface, application, Word

Description automatically generated

# Error detection and correction

**1**.Error using not the same name from mobile class for makeACall.

if(getDisplayNumber()!=-1&& gadget instanceof Mobile){

Mobile mobile= (Mobile) item.get(getDisplayNumber());

mobile.makeaCall(getDuration(),getPhoneNumber());

}

Fixed by changing the name to same ass mobile class makeACall.

**2**.Button not functioning

if DisplayAllButton = new JButton (“Display All”) not same as

if (command.equals("Display all")){

addMobile();

}

The button doesn’t do anything.

Fixed by matching the names, even one space more can lead to button not functioning BlueJ doesn’t give you error so if you don’t know is hard to find the problem.

**3**.Using wrong operator and variable in statement in add mp3 using wrong operator ==

f(getModel()==null||getPrice()==0 ||getWeight()==0||getSize()==null||getMemory()==0)

{

JOptionPane.showMessageDialog(frame,"A New Mp3 is not added ","error",

JOptionPane.ERROR\_MESSAGE);

}

else

{Mp3 newMp3= new Mp3 (getModel(),getPrice(),getWeight(), getSize(),getMemory());

item.add(newMp3);

JOptionPane.showMessageDialog(frame," A new Mp3 is added");

}

If enter minus amount to getPrice()==0 ||getWeight()==0||getSize()==null||getMemory()==0) fields display message pop-up

Graphical user interface, application

Description automatically generated

But Mp3 still added with negative number

Graphical user interface, application

Description automatically generated

Fixed by changing operators to (getModel()==null||getPrice()<=0 ||getWeight()<=0||getSize()==null||getMemory()<=0)

# Conclusion

Course work 2 was interesting and had much more depth into programming. As I started with a sketch how my program would look like button, text layout etc. The visual part of this course work is the longest as the code is quite long, and you have plenty of options to play around.

Combining course work 1 with this was not very smooth as the task tell us that we don’t need to change anything there was something I had to change even my course work 1 had no issue. Although it was only because I try to implement few extra buttons for better functionality.

Further more during course work 2 learned how ArrayList, casting and pop-up messages works also

basic visual programming which in my opinion was the most interesting part of CW2.

# APPENDIX: PROJECT CODE

**Gadget shop**

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import java.util.ArrayList;

import javafx.scene.control.Cell;

/\*\*

\* The class GadgetShop

\* Author (Andrius Uzkuraitis)

\* Version (18/04/2022)

\*

\*

\* GUI

\* Frame

\* Lables

\* Fields

\* Buttons

\*/

public class GadgetShop implements ActionListener

{

private JFrame frame;

private JLabel ModelLable;

private JLabel PriceLable;

private JLabel WeightLable;

private JLabel SizeLable;

private JLabel CreditLable;

private JLabel MemoryLable;

private JLabel PhoneNumber;

private JLabel Duration;

private JLabel Download;

private JLabel DisplayNumber;

private JLabel AddCredit;

private JLabel DeleteMusic;

private JTextField ModelTextbox;

private JTextField PriceTextbox;

private JTextField WeightTextbox;

private JTextField SizeTextbox;

private JTextField CreditTextbox;

private JTextField MemoryTextbox;

private JTextField PhoneNumberTextbox;

private JTextField DurationTextbox;

private JTextField DownloadTextbox;

private JTextField DisplayNumberTextbox;

private JTextField AddCreditTextbox;

private JTextField DeleteMusicTextbox;

private JButton addMobileButton;

private JButton addMp3Button;

private JButton ClearButton;

private JButton DisplayAllButton;

private JButton MakeAcallButton;

private JButton DownloadMusicButton;

private JButton AddCreditButton;

private JButton DeleteMusicButton;

ArrayList<Gadget> item= new ArrayList<Gadget>();

public GadgetShop ()

{

makeFrame();

}

public void makeFrame ()

{

frame = new JFrame("Gadget Shop");

Container ContentPane= frame.getContentPane();

ContentPane.setLayout(new BorderLayout(50,120));

ContentPane.setBackground(new Color(198,219,229));

frame.setSize(800,800);

//Top

JPanel panelNorth=new JPanel();

panelNorth.setLayout(new FlowLayout());

panelNorth.setBackground(new Color(198,219,229));

frame.getContentPane().add(panelNorth,"North");

JLabel GadgetShopLabel = new JLabel("Gadget Shop");

GadgetShopLabel.setFont(new Font("Serif", Font.BOLD,40 ));

GadgetShopLabel.setForeground(new Color(96,96,96));

panelNorth.add(GadgetShopLabel);

// West panel

JPanel panelWest=new JPanel();

panelWest.setLayout(new BoxLayout(panelWest,BoxLayout.Y\_AXIS));

panelWest.setBackground(new Color(198,219,229));

frame.getContentPane().add(panelWest,"West");

JLabel modelLable = new JLabel("Model ");

modelLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelWest.add(modelLable);

ModelTextbox = new JTextField(10);

panelWest.add(ModelTextbox);

JLabel priceLable = new JLabel("Price");

priceLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelWest.add(priceLable);

PriceTextbox = new JTextField(10);

panelWest.add(PriceTextbox);

JLabel weightLable = new JLabel("Weight");

weightLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelWest.add(weightLable);

WeightTextbox = new JTextField(10);

panelWest.add(WeightTextbox);

JLabel sizeLable = new JLabel("Size");

sizeLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelWest.add(sizeLable);

SizeTextbox = new JTextField(10);

panelWest.add(SizeTextbox);

JLabel creditLable = new JLabel("Credit");

creditLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelWest.add(creditLable);

CreditTextbox = new JTextField(10);

panelWest.add(CreditTextbox);

JLabel memoryLable = new JLabel("Memory");

memoryLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelWest.add(memoryLable);

MemoryTextbox = new JTextField(10);

panelWest.add(MemoryTextbox);

//Center panel for buttons

JPanel panelCenter=new JPanel();

panelCenter.setLayout(new GridLayout(4,4));

frame.getContentPane().add(panelCenter,"Center");

panelCenter.setBackground(new Color(198,219,229));

addMobileButton = new JButton("Add Mobile");

addMobileButton.setFont(new Font("Serif", Font.BOLD,25 ));

addMobileButton.addActionListener(this);

panelCenter.add(addMobileButton);

addMp3Button = new JButton("Add Mp3");

addMp3Button.setFont(new Font("Serif", Font.BOLD,25 ));

addMp3Button.addActionListener(this);

panelCenter.add(addMp3Button);

MakeAcallButton = new JButton("Make A Call");

MakeAcallButton.setFont(new Font("Serif", Font.BOLD,25 ));

MakeAcallButton.addActionListener(this);

panelCenter.add(MakeAcallButton);

DownloadMusicButton = new JButton("Download Music");

DownloadMusicButton.setFont(new Font("Serif", Font.BOLD,25 ));

DownloadMusicButton.addActionListener(this);

panelCenter.add(DownloadMusicButton);

AddCreditButton = new JButton("Add Credit");

AddCreditButton.setForeground(Color.BLUE);

AddCreditButton.setFont(new Font("Serif", Font.BOLD,25 ));

AddCreditButton.addActionListener(this);

panelCenter.add(AddCreditButton);

DeleteMusicButton = new JButton("Delete Music");

DeleteMusicButton.setForeground(Color.RED);

DeleteMusicButton.setFont(new Font("Serif", Font.BOLD,25 ));

DeleteMusicButton.addActionListener(this);

panelCenter.add(DeleteMusicButton);

//Eest panel

JPanel panelEast=new JPanel();

panelEast.setLayout(new BoxLayout(panelEast,BoxLayout.Y\_AXIS));

frame.getContentPane().add(panelEast,"East");

panelEast.setBackground(new Color(198,219,229));

JLabel PhoneNumberLable = new JLabel("Phone Number");

PhoneNumberLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelEast.add(PhoneNumberLable);

PhoneNumberTextbox = new JTextField(10);

panelEast.add(PhoneNumberTextbox);

JLabel DurationLable = new JLabel("Duration");

DurationLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelEast.add(DurationLable);

DurationTextbox = new JTextField(10);

panelEast.add(DurationTextbox);

JLabel DownloadLable = new JLabel("Download");

DownloadLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelEast.add(DownloadLable);

DownloadTextbox = new JTextField(10);

panelEast.add(DownloadTextbox);

JLabel DisplayNumberLable = new JLabel("Display Number");

DisplayNumberLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelEast.add(DisplayNumberLable);

DisplayNumberTextbox = new JTextField(10);

panelEast.add(DisplayNumberTextbox);

JLabel AddCreditLable = new JLabel("Add Credit");

AddCreditLable.setForeground(Color.BLUE);

AddCreditLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelEast.add(AddCreditLable);

AddCreditTextbox = new JTextField(10);

panelEast.add(AddCreditTextbox);

JLabel DeleteMusicLable = new JLabel("Delete Music");

DeleteMusicLable.setForeground(Color.RED);

DeleteMusicLable.setFont(new Font("Serif", Font.BOLD,25 ));

panelEast.add(DeleteMusicLable);

DeleteMusicTextbox = new JTextField(10);

panelEast.add(DeleteMusicTextbox);

//South Panel

JPanel panelSouth=new JPanel();

panelSouth.setLayout(new FlowLayout());

frame.getContentPane().add(panelSouth,"South");

panelSouth.setBackground(new Color(198,219,229));

DisplayAllButton = new JButton("Display All");

DisplayAllButton.setFont(new Font("Serif", Font.BOLD,25 ));

DisplayAllButton.addActionListener(this);

panelSouth.add(DisplayAllButton);

ClearButton = new JButton("Clear");

ClearButton.setForeground(Color.RED);

ClearButton.setFont(new Font("Serif", Font.BOLD,25 ));

ClearButton.addActionListener(this);

panelSouth.add(ClearButton);

frame.pack();

//Set Location

frame.setLocationRelativeTo(null);

//Set visible

frame.setVisible(true);

//Set size of the frame

frame.setResizable(false);

}

public static void main(String[] args)

{

GadgetShop Gadget = new GadgetShop();

}

/\*\*GUI

\* Functionality

\*/

public void actionPerformed(ActionEvent event)

{

String command = event.getActionCommand();

//Add Mobile

if (command.equals("Add Mobile")){

addMobile();

}

//Add Mp3

if (command.equals("Add Mp3")){

addMp3();

}

//Clear the fields

if(command.equals("Clear")){

clear();

}

//Make a Call

if(command.equals("Make A Call")){

makeACall();

}

//Display all gadgets

if(command.equals("Display All")){

DisplayAll();

}

//Downlead Music

if(command.equals("Download Music")){

DownloadMusic();

}

//Add credit

if(command.equals("Add Credit")){

addCredit();

}

// Delete Music

if(command.equals("Delete Music")){

DeleteMusic();

}

}

//Add Mobile

public void addMobile()

{

if (getModel()==null||getPrice()<=0 ||getWeight()<=0||getSize()==null||getCredit()<=0)

{

JOptionPane.showMessageDialog(frame, "A New Mobile is not added ","error",

JOptionPane.ERROR\_MESSAGE);

}

else

{Mobile newMobile= new Mobile (getModel(), getPrice(),getWeight(), getSize(),getCredit());

item.add(newMobile);

JOptionPane.showMessageDialog(frame," A new Mobile is added");

}

}

//Add Mp3

public void addMp3()

{

if(getModel()==null||getPrice()<=0 ||getWeight()<=0||getSize()==null||getMemory()<=0)

{

JOptionPane.showMessageDialog(frame,"A New Mp3 is not added ","error",

JOptionPane.ERROR\_MESSAGE);

}

else

{Mp3 newMp3= new Mp3 (getModel(),getPrice(),getWeight(), getSize(),getMemory());

item.add(newMp3);

JOptionPane.showMessageDialog(frame," A new Mp3 is added");

}

}

//Make a call

public void makeACall()

{

if (getPhoneNumber()==null||getDuration()<=0||getDisplayNumber()<0){

JOptionPane.showMessageDialog(frame,"Call is cancelled ","error",

JOptionPane.ERROR\_MESSAGE);

}

else{

try{

Gadget gadget = item.get(getDisplayNumber());

if(getDisplayNumber()!=-1&& gadget instanceof Mp3){

Mp3 mp3= (Mp3) item.get(getDisplayNumber());

mp3.DownloadMusic(getMusicFile());

}

else{

JOptionPane.showMessageDialog(frame,"This is not a Mp3");

}

}

catch(IndexOutOfBoundsException exception){

JOptionPane.showMessageDialog(frame,"Gadget with this number does not exist, Please check your entry.");

}

}

}

//Add Credit

public void addCredit()

{

if (getAmount()<=0||getDisplayNumber()<0){

JOptionPane.showMessageDialog(frame,"credit is Not added ","error",

JOptionPane.ERROR\_MESSAGE);

}

else{

try{

Gadget gadget = item.get(getDisplayNumber());

if(getDisplayNumber()!=-1&& gadget instanceof Mobile){

Mobile mobile= (Mobile) item.get(getDisplayNumber());

mobile.addCredit(getAmount());

}

else{

JOptionPane.showMessageDialog(frame,"This is not a Mobile");

}

}

catch(IndexOutOfBoundsException exception){

JOptionPane.showMessageDialog(frame,"Gadget with this number does not exist, Please check your entry.");

}

}

}

//Delete Music

public void DeleteMusic()

{

if (getamount()<=0||getDisplayNumber()<0){

JOptionPane.showMessageDialog(frame," Can't delete music ","error",

JOptionPane.ERROR\_MESSAGE);

}

else{

try{

Gadget gadget = item.get(getDisplayNumber());

if(getDisplayNumber()!=-1&& gadget instanceof Mp3){

Mp3 mp3= (Mp3) item.get(getDisplayNumber());

mp3.DeleteMusic(getamount());

}

else{

JOptionPane.showMessageDialog(frame,"This is not a Mp3");

}

}

catch(IndexOutOfBoundsException exception){

JOptionPane.showMessageDialog(frame,"Gadget with this number does not exist, Please check your entry.");

}

}

}

// add credit to Mobile

public int getAmount()

{

int theamount=0;

try{

theamount=Integer.parseInt(AddCreditTextbox.getText());

if(theamount<=0){

JOptionPane.showMessageDialog(frame,"Credit have to be positive or grater then 0");

}

}

catch(NumberFormatException exception) {

JOptionPane.showMessageDialog(frame,"Add credit field is empty or format is wrong");

}

return theamount;

}

// Delete music

public int getamount()

{

int theAmount=0;

try{

theAmount=Integer.parseInt(DeleteMusicTextbox.getText());

if(theAmount<=0){

JOptionPane.showMessageDialog(frame,"Delete music number have to be positive or grater then 0");

}

}

catch(NumberFormatException exception) {

JOptionPane.showMessageDialog(frame,"Delete music field is empty or format is wrong");

}

return theAmount;

}

public int numberOfGadgets()

{

return item.size();

}

//Download Music

public int getMusicFile()

{

int theMusicFile=0;

try{

theMusicFile=Integer.parseInt(DownloadTextbox.getText());

if(theMusicFile<=0){

JOptionPane.showMessageDialog(frame,"Download amount have to be positive or grater then 0");

}

}

catch(NumberFormatException exception) {

JOptionPane.showMessageDialog(frame,"Download field is empty");

}

return theMusicFile;

}

// gadget display number

public int getDisplayNumber()

{

int getDisplayNumber = -1;

try{

getDisplayNumber=Integer.parseInt(DisplayNumberTextbox.getText());

if(getDisplayNumber<0){

JOptionPane.showMessageDialog(frame,"Number is not valid, Please check your entry");

}

}

catch(NumberFormatException exception) {

JOptionPane.showMessageDialog(frame,"Display number field is empty or wrong format");

}

return getDisplayNumber;

}

// Mobile phone number

public String getPhoneNumber()

{

String thePhoneNumber=null;

try{

if(!(PhoneNumberTextbox.getText().length()==11)){

throw new Exception();

}

thePhoneNumber=PhoneNumberTextbox.getText();

}

catch(Exception e){

JOptionPane.showMessageDialog(frame,"phone number have to be 11 digits");

}

return thePhoneNumber;

}

//Duration of phone call

public int getDuration()

{ int theDuration=0;

try{

theDuration=Integer.parseInt(DurationTextbox.getText());

if(theDuration<=0){

JOptionPane.showMessageDialog(frame,"Duration amount have to be positive or grater then 0");

}

}

catch(NumberFormatException exception) {

JOptionPane.showMessageDialog(frame,"Duration field is empty");

}

return theDuration;

}

//Returns Model for gadget

public String getModel()

{

String aModel=null;

try{

if(ModelTextbox.getText().isEmpty()){

throw new Exception();

}

aModel=ModelTextbox.getText();

}

catch(Exception e){

JOptionPane.showMessageDialog(frame,"Model field is empty");

}

return aModel;

}

//Returns Price for gadget

public double getPrice()

{

double aPrice = 0;

try{

aPrice=Double.parseDouble(PriceTextbox.getText());

if(aPrice <=0){

JOptionPane.showMessageDialog(frame,"Price amount have to be positive or grater then 0");

}

}

catch(NumberFormatException exception) {

JOptionPane.showMessageDialog(frame, "Price field empty or price format is wrong");

}

return aPrice;

}

//Returns Weight for gadget

public int getWeight()

{

int aWeight = 0;

try{

aWeight=Integer.parseInt(WeightTextbox.getText());

if(aWeight <=0){

JOptionPane.showMessageDialog(frame,"Weight amount have to be positive or grater then 0");

}

}

catch(NumberFormatException exception) {

JOptionPane.showMessageDialog(frame, "Weight field empty or price format is wrong");

}

return aWeight;

}

//Returns Size for gadget

public String getSize()

{

String aSize=null;

try{

if(SizeTextbox.getText().isEmpty()){

throw new Exception();

}

aSize=ModelTextbox.getText();

}

catch(Exception e){

JOptionPane.showMessageDialog(frame,"Size field is empty");

}

return aSize;

}

//Returns credit for Mobile

public int getCredit()

{

int theCredit=0;

try{

theCredit=Integer.parseInt(CreditTextbox.getText());

if(theCredit <=0){

JOptionPane.showMessageDialog(frame,"Credit amount have to be positive or grater then 0");

}

}

catch(NumberFormatException exception){

JOptionPane.showMessageDialog(frame," Credit field empty or credit format is wrong.");

}

return theCredit;

}

// Returns memory for Mp3

public int getMemory()

{

int aMemory=0;

try{

aMemory=Integer.parseInt(MemoryTextbox.getText());

if(aMemory <=0){

JOptionPane.showMessageDialog(frame,"Memory amount have to be positive or grater then 0");

}

}

catch(NumberFormatException exception){

JOptionPane.showMessageDialog(frame," Memory field empty or memory format is wrong.");

}

return aMemory;

}

// Display all gadgets in ArrayList

public void DisplayAll()

{

clearTerminal();

for (Gadget gadget:item)

{

System.out.print((item.indexOf(gadget))+0+":");

gadget.display();

System.out.println();

}

}

// Clear all text fields in GUI

public void clear()

{

ModelTextbox.setText(null);

PriceTextbox.setText(null);

WeightTextbox.setText(null);

SizeTextbox.setText(null);

CreditTextbox.setText(null);

MemoryTextbox.setText(null);

PhoneNumberTextbox.setText(null);

DurationTextbox.setText(null);

DownloadTextbox.setText(null);

DisplayNumberTextbox.setText(null);

AddCreditTextbox.setText(null);

DeleteMusicTextbox.setText(null);

}

// Clear terminal

public void clearTerminal(){

System.out.print('\u000c');

}

}