**CMSC234**

**Project 4A**

**Lottery Game**

**Derya O. Kurin**

**Project Design:**

There are 4 activities in this lottery app: MainActivity, Pick4Activity, PowerBall Activity, and PlayActivity

**MainActivity:**

This activity contains 3 buttons for launching the Pick4Activity, PowerBall Activity, and PlayActivity.

**Pick4Activity:**

User starts with this activity to play Pick 4 game.

Quick Pick buttons help the user to fill the fields with random numbers between 0 and 9.

After the user fills 3 of the drawings, she can click purchase and play PowerBall to play PowerBall game.

Declare 4 EditText views for each each number of the 3 guesses.

Declare 3 Buttons for quick pick

Declare 1 Button for Purchasing and sending data through intent to PowerBall activity

Declare 3 Strings to hold 3 guesses as String characters.

Declare a pick4 String array to hold String representation of numbers from 0 to 9.

PROCESS: bindViews:void

No arg

This method binds all the variables with their corresponding view ids.

END PROCESS

onCreate:

Call bindView

Register eventlisteners to Buttons

For the first 3 quick pick buttons, declate methods to generate random numbers

For the purchase button, register it to purchaseAndPlayPB method

END PROCESS

PROCESS: generateRandom(): no arg:void

For generating randomness;

Using Collections, shuffle the pick4 String array

Register the first line of numbers to the first 4 elements of the array

PROCESS: saveAndPlayPB:no arg:void

Create and intent, to be sent the extras to PowerBallActivity

Get texts from guessed numbers for each line and store them in String variables pick4Line1, pick4Line2, pick4Line3.

Send these 3 Strings as Extra to PowerBall activity

Start intent

END PROCESS

**PowerBall Activity**

The guessed numbers in pick4 were sent to PowerBall before the user starts playing PowerBall game.

Declare String arrays for the randomness:

Declare a String[] powerBall to hold String representation of numbers from 1 to 26.

Declare a String[] rand5Number to hold String representation of numbers from 1 to 69.

Declare EditTexts for the 6 numbers for each 3 guesses.

Declare 3 Strings to hold Pick4 guessed numbers coming from the intent

Declare 3 Strings to hold PowerBall guessed numbers of this activity

PROCESS: onCreate

Call bindViews method

Declare a Bundle extras to get guessed pick4 numbers and store them in corresponding String variables: pickLine1, pickLine2, pickLine3

Register quick pick buttons with generateRandom method

Register Purchase button with purchase method

ENDPROCESS

PROCESS: generateRandom: void

No arg

Use Collections to shuffle the String array rand5ball for randomness

Use Collections to shuffle the String array powerball for randomness

Set text of first 5 EditText fields for numbers from 1st to the 5th element of the String array.

For the PowerBall set the last number’s text to 1st element of the powerball string array.

ENDPROCESS

PROCESS: purchase:void

No arg

Declare and intent to be sent to PlayActivity

Get text from EditViews and store them in pBLine1, pBLine2, pBLine3 that stands for the 3 guesses of the user for PowerBall game

Send these Strings and the Strings retrived from Pick4 Avtivity that holds user guesses for Pick4 numbers to PlayActivity

Start intent

ENDPROCESS

**PlayActivity:**

PlayActivity is the place where the winning numbers are generated and compared to the user’s guesses for Pick4 and powerBall games. It also shows if any of the tickets matches the winning numbers and shows how much the user won.

Declare a pick4 String array to hold String representation of numbers from 0 to 9.

Declare a String[] powerBall to hold String representation of numbers from 1 to 26.

Declare a String[] rand5Number to hold String representation of numbers from 1 to 69.

Declare EditText to hold winning numbers for Pick4 and powerBall

Declare EditText for how many tickets matched the winning numbers

Declare EditText to hold how much money the user won.

Declare 3 Strings for holding user guesses for Pick4

Declare 1 String to hold winning Pick4 number

Declare 3 Strings for holding user guesses for PowerBall

Declare 1 String to hold winning PowerBall number(First 5 white balls)

Declare 1 String to hold winning PowerBall number(Last red ball)

PROCESS: OnCreate

Register declared view variables with their corresponding ids

Retrieve the extras from Bundle intent and store them in Strings for user guessed numbers for pick4 and powerball games.

Call random4 method and store the returning value in String to hold the winning Pick4 number

Call random5Ball and store the returning value in String to hold the winning Powerball number( the first 5 white balls)

Call randomPBNo and store the returning value in String to hold the winning Powerball number( the last red ball)

Call winningTicketsP4 and store the returning value in int variable to hold the number of winning pick4 tickets

Set the corresponding view with winning Pick4 ticket number

Call winningTicketsPB and store the returning value in int variable to hold the number of winning powerBall tickets

Set the corresponding view with winning PowerBall ticket number

Call winAmount and store the returning double value in a double variable to hold the amount won by the user

Set the corresponding view with the amount won.

ENDPROCESS

PROCESS: rand5Ball: void

No arg

Use Collections to shuffle the String array rand5ball for randomness

Append the first 5 elements of rand5ball array to a String and return its value

ENDPROCESS

PROCESS: random4(): no arg:void

For generating randomness;

Using Collections, shuffle the pick4 String array

Append the first 4 elements of pick4 array to a String and return its value

ENDPROCESS

PROCESS: randomPBNo(): void

No arg

Use Collections to shuffle the String array powerball for randomness

Append the first element to a String and return its value

ENDPROCESS

PROCESS: WinningTicketsP4( String winningPick4, String p4Line1, String p4Line2, String p4Line3): int

Split winningPick4 by “” to each separate number and store it in an array wiiningP4

Repeat the same process for the three lines that stand for the String representation of the user’s guessed numbers

Call sort on Arrays class and pass each of the arrays above to have them sorted.

Declate and int variable: noOfTickets

Call equals to compare each guessed numbers with the winningP4 if they match.

Increment the noOfTickets if they match

Return the noOfTickets value

ENDPROCESS

PROCESS: winningTicketsPB(String winningPB, String pb, String pBLine1, String pBLine2, String pBLine3)

: int

Split winningPick4 by “ ” ( with a space char) to each separate number and store it in an array winning5ball

Repeat the same process for the three lines that stand for the String representation of the user’s guessed numbers (I splitted the String by “ “ since the way I append the guessed numbers to the Strings was with space characters between each number representation)

Store the last element of the arrays in separate String variables for the red ball

Call copyOf method on Arrays class and pass the arrays above to have new arrays without the last element in them(last element is red powerball).

Call sort method on Arrays class to sort the array for winning number and the arrays for guessed numbers.

Call if/else statement to check if they match using equals

If so increment noOfTickets

Return the noOfTickets

ENDPROCESS

PROCESS: winningAmount(int p4, int pB):int

Return the value of multiplication of p4 with 20000(prize) plus the multiplication of pB with 100000.

ENDPROCESS

PROCESS: bindView: void

No arg

Bind all the declared variables with their corresponding ids.

ENDPROCESS

**Lessons learned:**

I found this project useful since I was be able to work with intents across activities as I did in GameEmulator project. I could have make the UI more appealing using animations but the time is tight and I have an exam for another course tomorrow. Otherwise making the UI with some animation would look nice and would give the feeling of a real lottery app.

In addition, it was not in the instructions, but I would also add a new method to check whether any combination of the numbers would match, not only checking if 4 or 6 of the numbers method to give away the user small perks if not the full prize. I can later make this app look better for my portfolio.

I did not use Random class to generate Randomness but used a String shuffle technique instead. I could have used Random class and use bound to give some limits for the pick4 and powerBall numbers alternatively.

Overall, the assignment was helpful for me to work with passing data between intents in different forms. I preferred to send the data as Strings but it could have been dome as ArrayLists or in other forms. I think choosing the most efficient data structure while sending extras are essential part of the design when working between different activities.

**Screen Shots:**

A screenshot of a cell phone

Description automatically generated

Graphical user interface, application

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a cell phone

Description automatically generated