**Skeleton Program**

You will be receiving a skeleton program to research over the holidays. You should use the following pages to make notes on the program. Identify what it does, how it works, any errors and what improvements you would make.

**Explain the system, its purpose, what it does, data input, data returned**

|  |
| --- |
| **System - Play the game ‘Bingo’** |
| **Purpose – To entertain people** |
| **What it does – Displays the bingo card, calls out the numbers** |
| **Data input – Did you win (Y/N), Menu choice (what sort of game you want to play)** |
| **Data returned – Return the assigned number in Bingo Card, [a lot of other stuff is returned]** |
|  |

**Class Diagram**

|  |
| --- |
| https://lh3.googleusercontent.com/CxZTbqr0tUC2Vd73znffwB-TlcHpUpy7fFkUL8TjZHlvh2I_s9y6z_-zoAPaxNO0N4MN43mlhvArlwaYBF8wZlPe7X-8m8m0nE-TVT6MHMfKCeJF-Ckt0Nap4Hf_yHB4vXL7TI5Y |

**Explain any inheritance**

|  |
| --- |
| **Class BingoBonusCard Inherits Class BingoCard 🡪Generate numbers for the card and display the card** |
|  |
|  |

**Class**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |
| --- | --- |
| Identifier | What it does (inheritance, aggregation etc) |
| BingoGame | Composition aggregation with number machine + bingo card  *General Hint: New (composition) as (Assosciation)* |

**Properties**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Role | Validation | Scope (G/L) | What it does |
| numbers | Create instance of Number Machine |  |  |  |
| playerCard | Create instance of BingoCard |  |  |  |
| Won | Ask the user to enter whether they won or not (1 or 0) | On the original code, there is no validation | Local |  |
| Caller | Stores value of next number to be called | There is no validation | Local |  |
| choice | Ask the user what they want to do (game wise – i.e. quit, play normal game, play bonus game | If choice <> 0,1,2,3, there is an output message and the function keeps on looping until the choice entered is 0,1,2,3 | Local |  |

**Methods**

List the subroutines involved in the program, parameters, return type, type, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Type  (F/P) | Parameters | Return | What it does |
| Constructor (New) | P |  |  | Prints |
| Play Game | P |  |  | * playerCard is an instance of bingocard * Numbers called * Ask user if they won or not * Clears the console after each number * Displays the card with next number * Checks if they have won and then displays appropriate message   \*NORMAL GAME (only win with a full house) |
| Play Bonus Game | P |  |  | * playerCard is an instance of bingoBonuscard * Numbers called * Ask user if they won or not * Clears the console after each number * Displays the card with next number * Checks if they have won and then displays appropriate message – either how they won or if they stopped too early   \*EXTRA GAME(WIN with full house or two lines?) |
| Caller | F |  | /nothing | * Prints the next number to be called * If it is a special one then prints that instead (i.e. legs 11) |
| Menu | F |  | true | * Through the sub menuOptions, outputs the menu options * Reads the users choice |
| MenuOptions | P |  |  | * Outputs the menu options |

**Class**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |
| --- | --- |
| Identifier | What it does (inheritance, aggregation etc) |
| BingoCard |  |

**Properties**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Role | Validation | Scope (G/L) | What it does |
| numbers |  |  |  |  |
| base |  |  |  |  |
| Row1/Row2/Row2 |  |  |  |  |
| cardNumbers |  |  |  |  |
| row |  |  |  |  |

**Methods**

List the subroutines involved in the program, parameters, return type, type, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Type  (F/P) | Parameters | Return | What it does |
| New (Constructor) | P |  |  |  |
| AssignNumbers | F |  | cardNumbers | !Can someone please help! thanks |
| Displaycard | P |  |  | Prints the numbers onto the screen in three rows and 9 columns |
| AssignRowPlaces | F |  | row | Assign row numbers – if there are duplicates, a new number is called essentially |
| rearrange | F | dataset (Integer), Size (integer | dataSet | Re-Orders the numbers |
| GameOver | F (overridable) | calledNumbers (Integer), tail (integer) | matched | Checks how many of the numbers have actually been matched |

**Class**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |
| --- | --- |
| Identifier | What it does (inheritance, aggregation etc) |
| BingoBonusCard | Inherits bingoCard |

**Properties**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Role | Validation | Scope (G/L) | What it does |
| matched | Check if matched numbers is the full house |  | l |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Methods**

List the subroutines involved in the program, parameters, return type, type, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Type  (F/P) | Parameters | Return | What it does |
| GameOver | F (overrides) | calledNumbers, tail | 1 | Returns 1 if it is a full house else shares appropriate message of how many have been matched |
| FullHouse | F | calledNumbers, tail | matched | Checks if it is a full house |
|  |  |  |  |  |
|  |  |  |  |  |

**Class**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |
| --- | --- |
| Identifier | What it does (inheritance, aggregation etc) |
| NumberMachine | Generates the numbers |

**Properties**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Role | Validation | Scope (G/L) | What it does |
| numberOrder | Rearrange the number order of the array |  | G |  |
| Current | Counter-ish |  | G |  |
| back | Counter -ish |  | G |  |
| temp | Help with the random order |  | L |  |
| Num1, Num2 | Help with the random order |  | l |  |

**Methods**

List the subroutines involved in the program, parameters, return type, type, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Type  (F/P) | Parameters | Return | What it does |
| New (Constructor) | P |  |  | Get numbers 1-75 and stores them in a random order in the NumberOrder array |
| Practice Game | P |  |  | Stores the numbers of the set array of numbers for the practice game |
| nextBall | f |  | numberOrder(Current-1) OR -1 | Goes through and returns one of the 74/75 numbers to be called or returns -1 depending on position in the array |
| getNumbers | F |  | numberOrder | Returns the number order array |
| getBack | F |  | current | Returns the current position it is at in the array |

**Class**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |
| --- | --- |
| Identifier | What it does (inheritance, aggregation etc) |
| repo | Static function – can be called anywhere without being instantiated |

**Properties**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Role | Validation | Scope (G/L) | What it does |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Methods**

List the subroutines involved in the program, parameters, return type, type, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Type  (F/P) | Parameters | Return | What it does |
| newRandom | F | Lowerbound,upperbound | (something complicated) | Helps generate a random number within a range |
|  |  | *Return CInt(Math.Floor((upperbound - lowerbound + 1) \* Rnd())) + lowerbound* | ->^ |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Errors**

|  |  |  |
| --- | --- | --- |
| **Where** | **Why** | **Solution** |
| Does it actually check the four corners in bingo bonus? |  |  |
| Does bingo bonus want 2 horizontal lines or just one? (the code says – well done two lines?) |  |  |
| Practice menu keeps repeating the menu display when selected |  |  |
| Allignment |  |  |
| Loops to 75 not 90 |  |  |
| Practice game numbers in wrong order |  |  |
| In all classes – uses dim instead of private for all properties |  |  |
| Numbers don’t reset when played once and want to play another |  |  |
| Good bye user |  |  |

**Improvements**

|  |  |  |
| --- | --- | --- |
| **Where** | **Why** | **Solution** |
| Display all numbers gone through |  |  |
| Try catch data entry |  |  |
| Use special names - duck |  |  |
| The first time a number is rolled says ‘and next number is…’ |  |  |
| Have a tally system to see how many times you have lost and how many times you have won |  |  |
| Show numbers previous called |  |  |
| Add option to show user the rules and how to play |  |  |
| Format card |  |  |
| Visually cross out the card |  |  |