KAI LIN CHANG

Oct 12, 2015

CS210, 4:00 – 5:15 Tue/Thur.

Professor Bob Wilson

Lab 3

**import** java.util.\*;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* This class extends from ArrayList to keep BingoBall

\* **@author** Kai Lin Chang

\* **@version** 1.0

\* Date: Oct. 11, 2015

\*

\* **@param** <T> T is generice type

\* **@param** rand rand is a radom object

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** ArraySet<T> **extends** ArrayList<T>{

**private** Random rand;

/\*\*

\* This constructs its super class and

\* instantiate a random object

\*/

**public** ArraySet(){

**super**();

rand = **new** Random();

}

/\*\*

\* This is to check the element being added,

\* return true if the element exists;

\* otherwise, return false

\*/

**public** **boolean** add(T elememt){

**if**(!**super**.contains(elememt))

**return** **true**;

**return** **false**;

}

/\*\*

\* This method is to remove an index from ArraySet

\* **@param** size the size of ArrarSet

\* **@return** return an element being removed from ArraySet

\*/

**public** T removeRandom(**int** size){

**return** **super**.remove(rand.nextInt(size - 1));

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* This is to run a Bingo Game

\* **@author** Kai Lin Chang

\* **@version** 1.0

\* Date: Oct. 11, 2015

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** BingoGame {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

ArraySet<BingoBall> myBingo = **new** ArraySet<BingoBall>();

**for** (**int** i = 0; i < 75; i++){

myBingo.add(i, **new** BingoBall(i+1));

}

BingoCard myBingoCard = **new** BingoCard();

**while**(**true**){

**if** (myBingoCard.cover(myBingo.removeRandom(myBingo.size())) && myBingoCard.hasBingo()){

System.***out***.println("Bingo!");

**break**;

}

}

System.***out***.print(myBingoCard.toString());

}

}

|  |
| --- |
|  |

1. **Does the project as initially set up compile?  If so, can you run it?  Record your observations for your lab report.**

Each class can compile, but the project cannot run because of no main method in this program.

1. **Explain why your project would compile successfully before you wrote the new BingoGame.java file.  Even if it did compile, could you run it?  Explain.**

Each class can compile, but the project cannot run because of no main method in this program.

1. **Did you notice the incremental development model that we have followed in these three labs?  We developed three versions of the program - each successively closer to the final goal. What has been good about this method?  What has been bad about it?**

Compared to the lab2, the lab3 is a more efficient method to develop a bingo game. The lab2 uses the certain number to check if there is a bingo or not. However, this does not make sense because the bingo ball should be picked randomly for each time, and once the number is picked, the number has to be removed. Hence, the lab3 implements this idea. Furthermore, lab3 uses generic approach that makes the whole program to be more efficient rather than lab2.