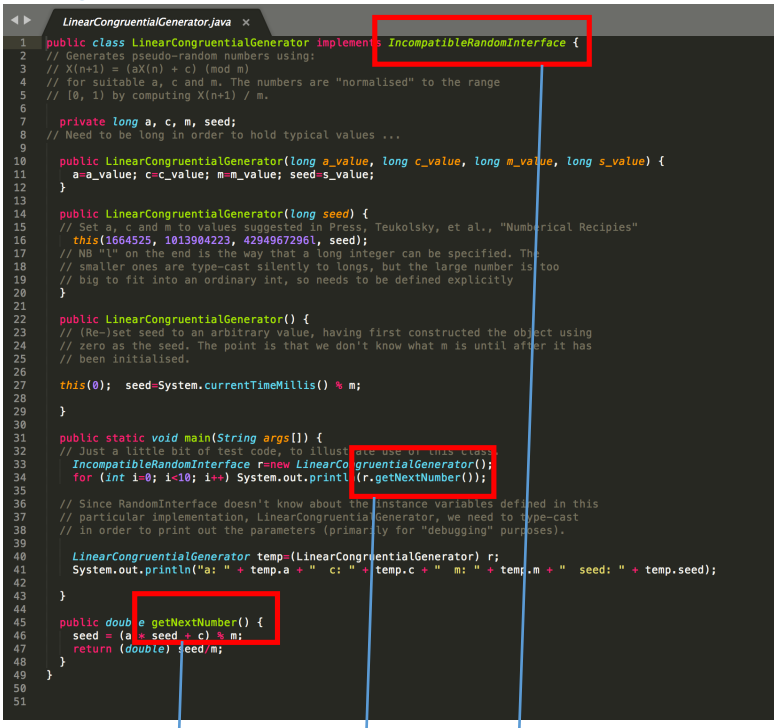
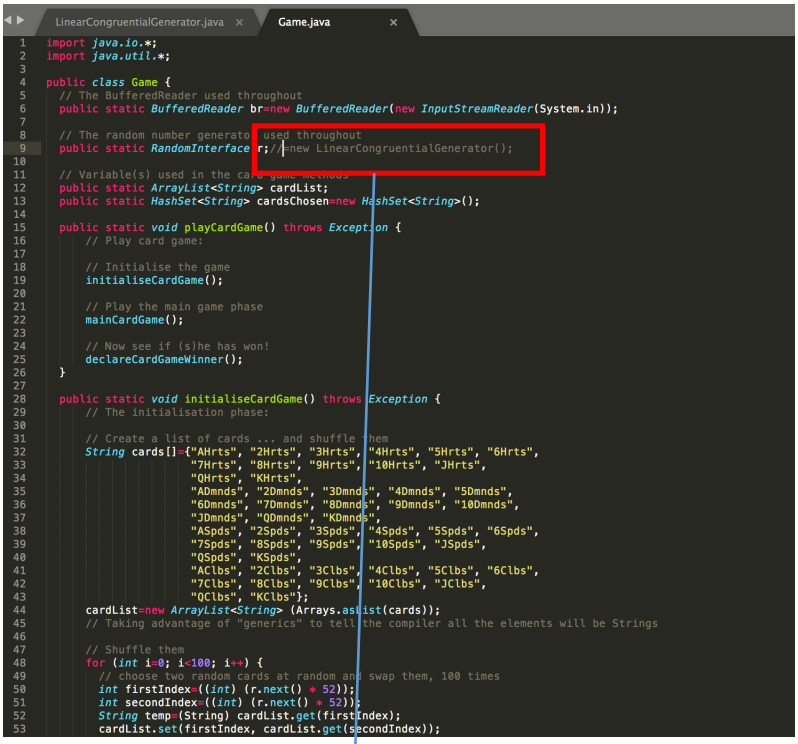
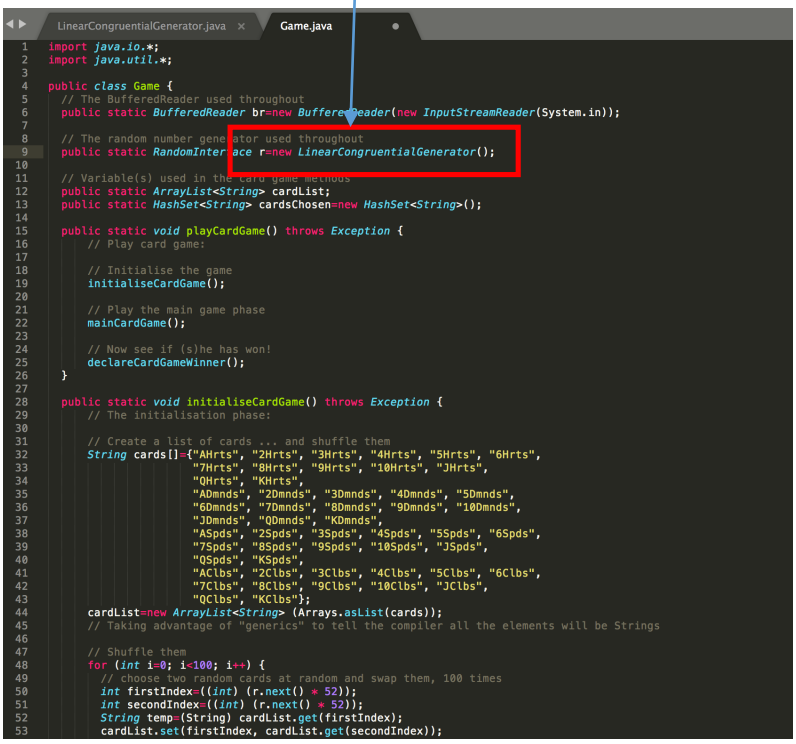
Question 1:

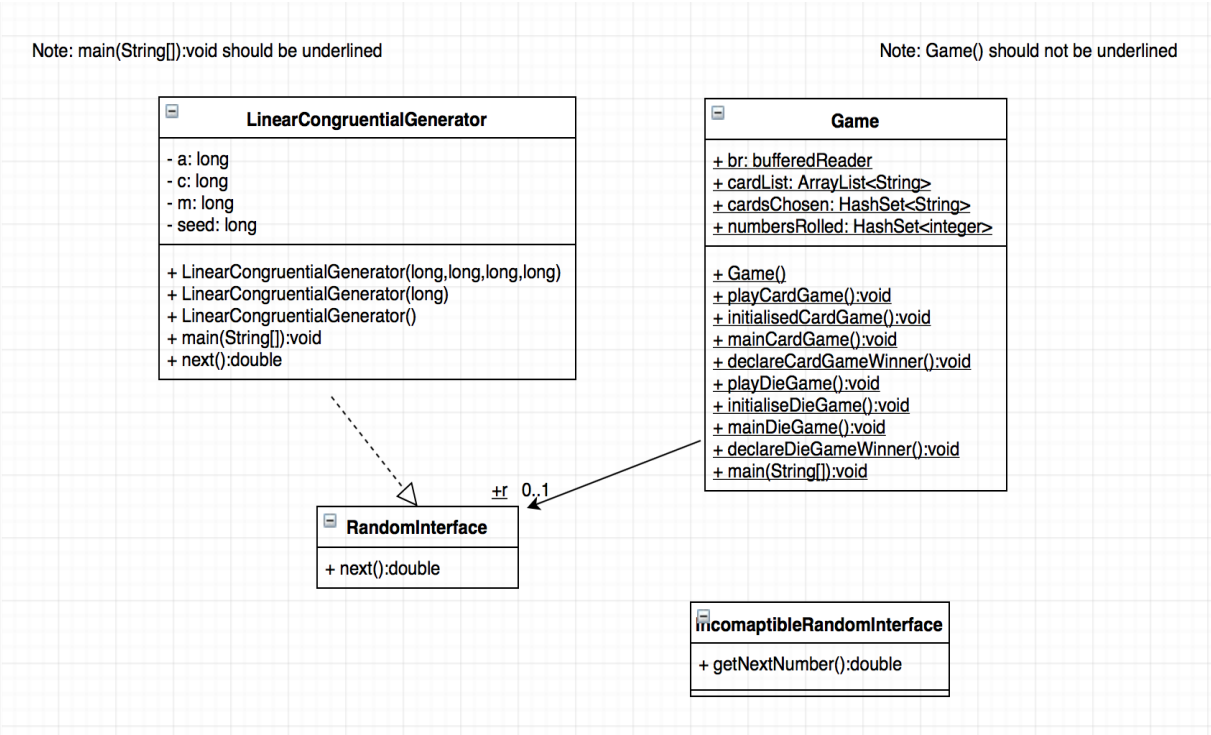
Changes Made:




LinearCongruentialGenerator.java
Game.java
import java.io.*;
import java.util.*;
public class Game {
// The BufferedReader used throughout
public static BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
// The random number generator used throughout
public static RandomInterface r;//=new LinearCongruentialGenerator();
// Variable(s) used in the card game methods
public static ArrayList<String> cardList;
public static HashSet<String> cardsChosen=new HashSet<String>();
public static void playCardGame() throws Exception {
// Play card game:
// Initialise the game
initialiseCardGame();
// Play the main game phase
mainCardGame();
// Now see if (s)he has won!
declareCardGameWinner();
}
public static void initialiseCardGame() throws Exception {
// The initialisation phase:
// Create a list of cards ... and shuffle them
String cards[]={"AHrts", "2Hrts", "3Hrts", "4Hrts", "5Hrts", "6Hrts",
"7Hrts", "8Hrts", "9Hrts", "10Hrts", "JHrts",
"QHrts", "KHrts",
"ADmnds", "2Dmnds", "3Dmnds", "4Dmnds", "5Dmnds",
"6Dmnds", "7Dmnds", "8Dmnds", "9Dmnds", "10Dmnds",
"JDmnds", "QDmnds", "KDmnds",
"ASpds", "2Spds", "3Spds", "4Spds", "5Spds", "6Spds",
"7Spds", "8Spds", "9Spds", "10Spds", "JSpds",
"QSpds", "KSpds",
"AClbs", "2Clbs", "3Clbs", "4Clbs", "5Clbs", "6Clbs",
"7Clbs", "8Clbs", "9Clbs", "10Clbs", "JClbs",
"QClbs", "KClbs"};
cardList=new ArrayList<String> (Arrays.asList(cards));
// Taking advantage of "generics" to tell the compiler all the elements will be Strings
// Shuffle them
for (int i=0; i<100; i++) {
// choose two random cards at random and swap them, 100 times
int firstIndex=((int) (r.next() * 52));
int secondIndex=((int) (r.next() * 52));
String temp=(String) cardList.get(firstIndex);
cardList.set(firstIndex, cardList.get(secondIndex));


LinearCongruentialGenerator.java
Game.java
import java.io.*;
import java.util.*;
public class Game {
// The BufferedReader used throughout
public static BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
// The random number generator used throughout
public static RandomInterface r=new LinearCongruentialGenerator();
// Variable(s) used in the card game methods
public static ArrayList<String> cardList;
public static HashSet<String> cardsChosen=new HashSet<String>();
public static void playCardGame() throws Exception {
// Play card game:
// Initialise the game
initialiseCardGame();
// Play the main game phase
mainCardGame();
// Now see if (s)he has won!
declareCardGameWinner();
}
public static void initialiseCardGame() throws Exception {
// The initialisation phase:
// Create a list of cards ... and shuffle them
String cards[]={"AHrts", "2Hrts", "3Hrts", "4Hrts", "5Hrts", "6Hrts",
"7Hrts", "8Hrts", "9Hrts", "10Hrts", "JHrts",
"QHrts", "KHrts",
"ADmnds", "2Dmnds", "3Dmnds", "4Dmnds", "5Dmnds",
"6Dmnds", "7Dmnds", "8Dmnds", "9Dmnds", "10Dmnds",
"JDmnds", "QDmnds", "KDmnds",
"ASpds", "2Spds", "3Spds", "4Spds", "5Spds", "6Spds",
"7Spds", "8Spds", "9Spds", "10Spds", "JSpds",
"QSpds", "KSpds",
"AClbs", "2Clbs", "3Clbs", "4Clbs", "5Clbs", "6Clbs",
"7Clbs", "8Clbs", "9Clbs", "10Clbs", "JClbs",
"QClbs", "KClbs"};
cardList=new ArrayList<String> (Arrays.asList(cards));
// Taking advantage of "generics" to tell the compiler all the elements will be Strings
// Shuffle them
for (int i=0; i<100; i++) {
// choose two random cards at random and swap them, 100 times
int firstIndex=((int) (r.next() * 52));
int secondIndex=((int) (r.next() * 52));
String temp=(String) cardList.get(firstIndex);
cardList.set(firstIndex, cardList.get(secondIndex));

UML Diagram:



Description:

The purpose of this program is to play a game. There are two games, a die game and and a card game. This is decided by the user inputting either a ‘c’ for card game or ‘d’ for die game. If the user inputs ‘c’ then card game is initialized and an array of cards are shown with the command “Hit <Return> to choose a card”. Once return is pressed you start playing the main card game and a random card is chosen and by pressing return again you get another random card. In order to win the game at least one of the random cards needs to be an Ace, otherwise you lose. If the user inputs ‘d’ then die game is initialized and the main die game begins with “Hit <Return> to role the die”. A random number is given between 1-6 and then press return again to receive another number. To win the game you must have received at least one ‘1’.

Question 2:

Improvements:

public interface Game {
Create Game interface to make the interface for the Factory Design.

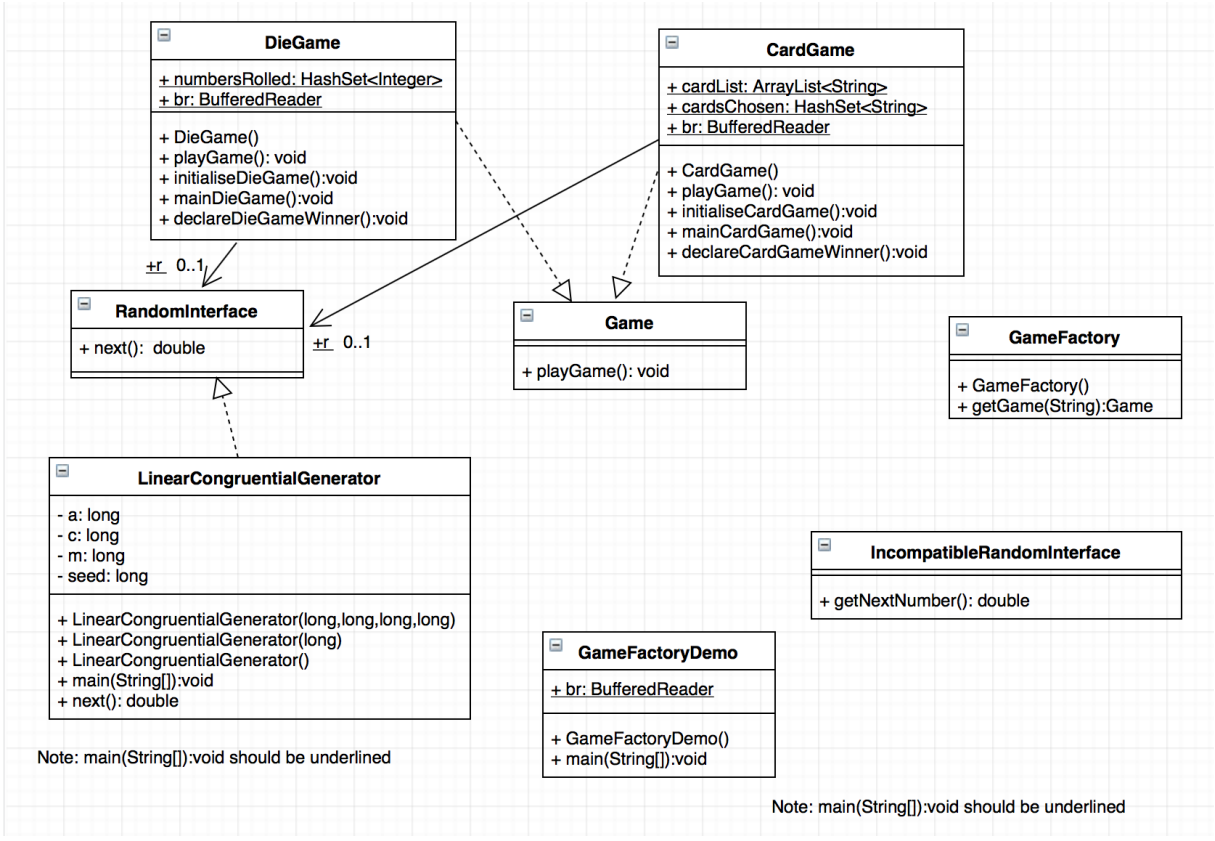
public class DieGame implements Game {
public class CardGame implements Game {
Create two new concrete classes DieGame and CardGame implementing the same Game interface. Both of which contain the code to play their game.

public class GameFactory {
Create a new class GameFactory that is a factory to generate game of concrete class based on information entered by user.

public class GameFactoryDemo {
Create a new class GameFactoryDemo that is used to get object of concrete class by passing an information such as type.

I chose a factory design as the pattern defines an interface for creating an object, but lets subclasses decide which class to instantiate. It removes instantiation of actual implementation classes from client code. The factory design will make my code more robust, less coupled and easy to extend.

UML Diagram:



Playing Game:

run GameFactoryDemo

Card Game

```
You chose 10Hrts
Hit <RETURN> to choose a card

You chose KClbs
Cards chosen: [KClbs, 10Hrts]
Remaining cards: [KDmnds, 8Hrts, 5Dmnds, AHrts, 4Hrts, 9Dmnds, 6Spds, JClbs, 3Dmnds, 3Clbs, JSpds, 7Dmnds, 6Hrts, ASpds, AClbs, 6Clbs, 2Hrts, KHrts, 10Clbs, 3Spds, KSpds, 2Dmnds, 5Clbs, 9Spds, QHrts, 10Dmnds, 6D
mnds, 3Hrts, JHrts, 9Hrts, JDmnds, 2Clbs, QClbs, 9Clbs, 4Dmnds, ADmnds, 8Clbs, 10Spds, 8Spds, QSpds, 4Spds, 5Hrts, QDmnds, 8Dmnds, 5Spds, 7Hrts, 2Spds, 4Clbs, 7Spds, 7Clbs]
Cards chosen: [KClbs, 10Hrts]
You lost!
lauras-MacBook-Pro-3:Source1 laura$ java GameFactoryDemo
Card (c) or Die (d) game? c
[KClbs, 8Spds, 3Dmnds, 5Dmnds, JSpds, 8Hrts, 4Dmnds, 10Dmnds, 9Spds, 6Hrts, JHrts, 5Hrts, AHrts, 4Clbs, ADmnds, 6Dmnds, 5Spds, AClbs, QSpds, 6Clbs, 3Spds, 3Clbs, 4Spds, 2Clbs, JClbs, 10Clbs, 7Hrts, KHrts, KDmnds
, 2Dmnds, 4Hrts, 6Spds, JDmnds, 9Clbs, 3Hrts, 7Dmnds, QClbs, ASpds, 8Dmnds, 9Dmnds, 7Clbs, KSpds, QDmnds, 7Spds, 9Hrts, 8Clbs, 2Spds, QHrts, 10Hrts, 5Clbs, 10Spds, 2Hrts]
Hit <RETURN> to choose a card

You chose 2Spds
Hit <RETURN> to choose a card

You chose 4Dmnds
Cards chosen: [2Spds, 4Dmnds]
Remaining cards: [KClbs, 8Spds, 3Dmnds, 5Dmnds, JSpds, 8Hrts, 10Dmnds, 9Spds, 6Hrts, JHrts, 5Hrts, AHrts, 4Clbs, ADmnds, 6Dmnds, 5Spds, AClbs, QSpds, 6Clbs, 3Spds, 3Clbs, 4Spds, 2Clbs, JClbs, 10Clbs, 7Hrts, KHrt
s, KDmnds, 2Dmnds, 4Hrts, 6Spds, JDmnds, 9Clbs, 3Hrts, 7Dmnds, QClbs, ASpds, 8Dmnds, 9Dmnds, 7Clbs, KSpds, QDmnds, 7Spds, 9Hrts, 8Clbs, QHrts, 10Hrts, 5Clbs, 10Spds, 2Hrts]
Cards chosen: [2Spds, 4Dmnds]
You lost!
lauras-MacBook-Pro-3:Source1 laura$ java GameFactoryDemo
Card (c) or Die (d) game? c
[7Clbs, AHrts, 8Clbs, JSpds, 5Spds, 2Spds, 8Spds, 10Hrts, JHrts, 5Dmnds, 2Hrts, 10Clbs, QSpds, 6Hrts, 7Spds, 4Spds, 2Clbs, AClbs, 6Dmnds, 6Clbs, 5Hrts, QHrts, KClbs, 4Clbs, 9Spds, 3Spds, KHrts, 10Dmnds, 3Clbs, 3
Hrts, 9Hrts, JDmnds, 2Dmnds, 4Hrts, 10Spds, QClbs, QDmnds, JClbs, 9Clbs, 4Dmnds, 6Spds, 7Hrts, 3Dmnds, ADmnds, 9Dmnds, 8Hrts, 8Dmnds, 5Clbs, 7Dmnds, KSpds, KDmnds, ASpds]
Hit <RETURN> to choose a card

You chose 5Hrts
Hit <RETURN> to choose a card

You chose 9Dmnds
Cards chosen: [5Hrts, 9Dmnds]
Remaining cards: [7Clbs, AHrts, 8Clbs, JSpds, 5Spds, 2Spds, 8Spds, 10Hrts, JHrts, 5Dmnds, 2Hrts, 10Clbs, QSpds, 6Hrts, 7Spds, 4Spds, 2Clbs, AClbs, 6Dmnds, 6Clbs, QHrts, KClbs, 4Clbs, 9Spds, 3Spds, KHrts, 10Dmnds
, 3Clbs, 3Hrts, 9Hrts, JDmnds, 2Dmnds, 4Hrts, 10Spds, QClbs, QDmnds, JClbs, 9Clbs, 4Dmnds, 6Spds, 7Hrts, 3Dmnds, ADmnds, 8Hrts, 8Dmnds, 5Clbs, 7Dmnds, KSpds, KDmnds, ASpds]
Cards chosen: [5Hrts, 9Dmnds]
You lost!
lauras-MacBook-Pro-3:Source1 laura$ java GameFactoryDemo
Card (c) or Die (d) game? c
[7Hrts, KSpds, JSpds, 7Dmnds, 8Spds, 4Dmnds, 3Hrts, KClbs, 8Clbs, KHrts, JClbs, 10Spds, 4Spds, 9Spds, 8Hrts, JHrts, 8Dmnds, 6Spds, 10Hrts, KDmnds, QSpds, 3Clbs, 2Spds, AClbs, QDmnds, 9Clbs, 5Spds, QClbs, 6Dmnds,
 6Hrts, 10Clbs, 2Hrts, 10Dmnds, AHrts, 7Clbs, JDmnds, 4Clbs, 4Hrts, 5Hrts, 7Spds, 3Dmnds, QHrts, ADmnds, 5Clbs, 3Spds, 2Clbs, ASpds, 6Clbs, 2Dmnds, 9Dmnds, 5Dmnds, 9Hrts]
Hit <RETURN> to choose a card

You chose 2Dmnds
Hit <RETURN> to choose a card

You chose ADmnds
Cards chosen: [2Dmnds, ADmnds]
Remaining cards: [7Hrts, KSpds, JSpds, 7Dmnds, 8Spds, 4Dmnds, 3Hrts, KClbs, 8Clbs, KHrts, JClbs, 10Spds, 4Spds, 9Spds, 8Hrts, JHrts, 8Dmnds, 6Spds, 10Hrts, KDmnds, QSpds, 3Clbs, 2Spds, AClbs, QDmnds, 9Clbs, 5Spd
s, QClbs, 6Dmnds, 6Hrts, 10Clbs, 2Hrts, 10Dmnds, AHrts, 7Clbs, JDmnds, 4Clbs, 4Hrts, 5Hrts, 7Spds, 3Dmnds, QHrts, 5Clbs, 3Spds, 2Clbs, ASpds, 6Clbs, 9Dmnds, 5Dmnds, 9Hrts]
Cards chosen: [2Dmnds, ADmnds]
You won!
lauras-MacBook-Pro-3:Source1 laura$
```

Die Game

```
You lost!
lauras-MacBook-Pro-3:Source1 laura$ java GameFactoryDemo
Card (c) or Die (d) game? d
Hit <RETURN> to roll the die

You rolled 6
Hit <RETURN> to roll the die

You rolled 2
Numbers rolled: [2, 6]
You lost!
lauras-MacBook-Pro-3:Source1 laura$ java GameFactoryDemo
Card (c) or Die (d) game? d
Hit <RETURN> to roll the die

You rolled 5
Hit <RETURN> to roll the die

You rolled 2
Numbers rolled: [2, 5]
You lost!
lauras-MacBook-Pro-3:Source1 laura$ java GameFactoryDemo
Card (c) or Die (d) game? d
Hit <RETURN> to roll the die

You rolled 5
Hit <RETURN> to roll the die

You rolled 6
Numbers rolled: [5, 6]
You lost!
lauras-MacBook-Pro-3:Source1 laura$ java GameFactoryDemo
Card (c) or Die (d) game?
Exception in thread "main" java.lang.Exception: Not Allowed
        at GameFactoryDemo.main(GameFactoryDemo.java:37)
lauras-MacBook-Pro-3:Source1 laura$ java GameFactoryDemo
Card (c) or Die (d) game? d
Hit <RETURN> to roll the die

You rolled 6
Hit <RETURN> to roll the die

You rolled 4
Numbers rolled: [4, 6]
You lost!
lauras-MacBook-Pro-3:Source1 laura$ java GameFactoryDemo
Card (c) or Die (d) game? d
Hit <RETURN> to roll the die

You rolled 1
Hit <RETURN> to roll the die

You rolled 4
Numbers rolled: [1, 4]
You won!
lauras-MacBook-Pro-3:Source1 laura$
```