Objectives

- Reviewing the semester
- Picasso demo, discussion

- Reminder: course evaluations due Sunday ~2 p.m.

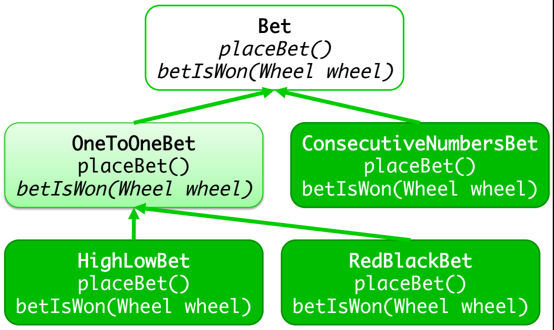
ASSIGNMENT 11 DISCUSSION

Review: Game class

```
private String placeBet(int whichBet) {
   String result = "";

   if (whichBet == 0) {
        Set<String> choices = new TreeSet<String>();
        choices.add(Wheel.BLACK);
        choices.add(Wheel.RED);
        result = ConsoleReader.promptOneOf("Please bet",
        choices);
   } else if (whichBet == 1) {
        Set<String> choices = new TreeSet<String>();
        choices.add("even");
        choices.add("odd");
        result = ConsoleReader.promptOneOf("Please bet",
        choices);
   } else if (whichBet == 2) {
        …
   }
   System.out.println();
   return result;
}
```

Hierarchy of Bets



Effect on Game Class

Greatly reduces amount of code in Game

```
private Bet[] mvPossibleBets = {
    new
    new
    new
    new
    new
    new
};
```

```
public void playRound() {
    …
    int whichBet = promptForBet();

    Bet betMade = myPossibleBets[whichBet];
    betMade.placeBet();

    spinWheel();

    if (betMade.betIsWon(myWheel)) {
        amount *= myPossibleBets[whichBet].getOdds();
    } else {
        amount *= -1;
    }

    player.updateBankroll(amount);
}
```

A Look at the Bet classes

```
public abstract class OneToOneBet extends Bet {

  protected Set<String> choices = new TreeSet<String>();

  public OneToOneBet(String description) {
       super(description, 1);
  }

  @Override
  public void placeBet() {
       userChoice = ConsoleReader.promptOneOf("Please bet",
               choices);
  }
}
```

A Look at the Bet classes

```
public HighLowBet() {
    super("High or Low");
    choices.add(HIGH);
    choices.add(LOW);
}

@Override
public boolean betIsWon(Wheel wheel) {
    if (wheel.onGreen()) {
        return false;
    }

    int wheelNumber = wheel.getNumber();

    return (wheelNumber > SEP && userChoice.equals(HIGH))
        || (wheelNumber <= SEP && userChoice.equals(LOW));

}
```

A Look at the Bet classes

```
public class ConsecutiveBet extends Bet {

    private int numConsecutive;

    public ConsecutiveBet(int numConsecutive, int odds) {
        super(numConsecutive + " in a row", odds);
        this.numConsecutive = numConsecutive;
    }

    public ConsecutiveBet(int odds) {
        this(1, odds);
    }

    public void placeBet() {
        userChoice = ""
        + ConsoleReader.promptRange("Enter first of " + numConsecutive
        + " consecutive numbers", 1, 34);
    }

    public boolean betIsWon(Wheel wheel) {
        int start = Integer.parseInt(userChoice);
        return (start <= wheel.getNumber() && wheel.getNumber() <
                start + this.numConsecutive);
    }
```

Effect on Game Class

Greatly reduces amount of code in Game

```
private Bet[] myPossibleBets = {
    new
    new
    new
    new
    new
    new
};
```

```
public void playRound() {
    …
    int whichBet = promptForBet();

    Bet betMade = myPossibleBets[whichBet];
    betMade.placeBet();

    spinWheel();

    if (betMade.betIsWon(myWheel)) {
        amount *= myPossibleBets[whichBet].getOdds();
    } else {
        amount *= -1;
    }

    player.updateBankroll(amount);
}
```

Discussion

- Benefits of the refactored hierarchy
- Drawbacks of the refactored hierarchy

Benefits of The Refactored Hierarchy

- Benefits of the refactored hierarchy
  - Where is the logic about the bets?
    - In the Bet classes
    - Game can manage the game, not be responsible for bets
  - Easier to add a new Bet
- Drawbacks of the refactored hierarchy
  - Adds more classes, hierarchy, abstraction

Oh, the places you have been!

- What Have You Learned This Semester?

Summary of Java Platform SE 6.0

Remember from the first day of class?

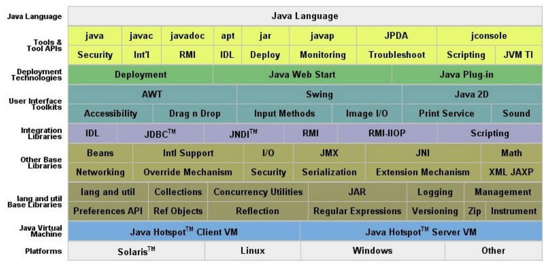
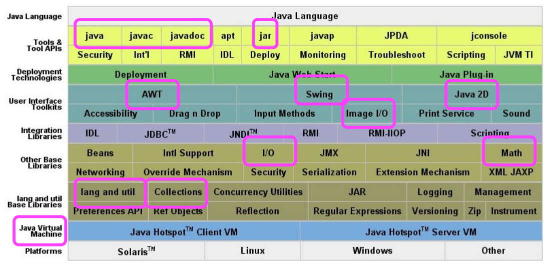

Java Language | Java Language
Tools & Tool APIs | java | javac | javadoc | apt | jar | javap | JPDA | jconsole
Security | Int'l | RMI | IDL | Deploy | Monitoring | Troubleshoot | Scripting | JVM TI
Deployment Technologies | Deployment | Java Web Start | Java Plug-in
User Interface Toolkits | AWT | Swing | Java 2D
Accessibility | Drag n Drop | Input Methods | Image I/O | Print Service | Sound
Integration Libraries | IDL | JDBC™ | JNDI™ | RMI | RMI-IIOP | Scripting
Other Base Libraries | Beans | Intl Support | I/O | JMX | JNI | Math
Networking | Override Mechanism | Security | Serialization | Extension Mechanism | XML JAXP
lang and util Base Libraries | lang and util | Collections | Concurrency Utilities | JAR | Logging | Management
Preferences API | Ref Objects | Reflection | Regular Expressions | Versioning | Zip | Instrument
Java Virtual Machine | Java Hotspot™ Client VM | Java Hotspot™ Server VM
Platforms | Solaris™ | Linux | Windows | Other

Image from Sun's site

Dec 11, 2009 Sprenkle - CS209 13

Summary of Java Platform SE 6.0

Remember from the first day of class?


Java Language | Java Language
Tools & Tool APIs | java | javac | javadoc | apt | jar | javap | JPDA | jconsole
Security | Int'l | RMI | IDL | Deploy | Monitoring | Troubleshoot | Scripting | JVM TI
Deployment Technologies | Deployment | Java Web Start | Java Plug-in
User Interface Toolkits | AWT | Swing | Java 2D
Accessibility | Drag n Drop | Input Methods | Image I/O | Print Service | Sound
Integration Libraries | IDL | JDBC™ | JNDI™ | RMI | RMI-IIOP | Scripting
Other Base Libraries | Beans | Intl Support | I/O | JMX | JNI | Math
Networking | Override Mechanism | Security | Serialization | Extension Mechanism | XML JAXP
lang and util Base Libraries | lang and util | Collections | Concurrency Utilities | JAR | Logging | Management
Preferences API | Ref Objects | Reflection | Regular Expressions | Versioning | Zip | Instrument
Java Virtual Machine | Java Hotspot™ Client VM | Java Hotspot™ Server VM
Platforms | Solaris™ | Linux | Windows | Other

Dec 11, 2009 Sprenkle - CS209 14

Project Notes

- Project Analysis: Make sure you understand the others' design/code/parts
  - *At least* at a high level

Dec 11, 2009 Sprenkle - CS209 15

PICASSO DEMO

Dec 11, 2009 Sprenkle - CS209 16

Extensions Discussion

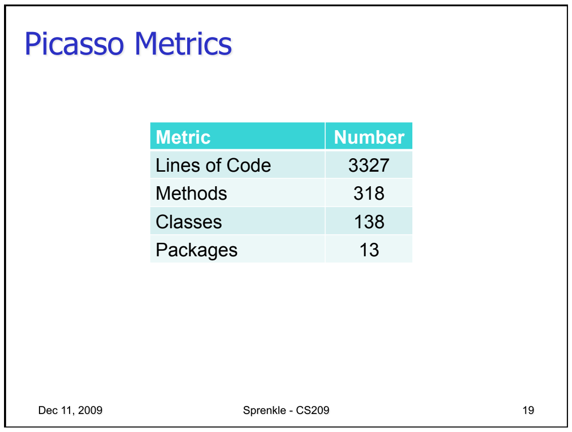
- What would you need to do to create random expressions?

Dec 11, 2009 Sprenkle - CS209 17

Picasso Demo

- Let's see this baby in action!
- Discuss any design issues/challenges you've met so far
  - Interesting discussions/conclusions
  - How you'd change if you were to do something similar later

Dec 11, 2009 Sprenkle - CS209 18



Picasso Metrics

Metric	Number
Lines of Code	3327
Methods	318
Classes	138
Packages	13