Java Workshop CSI W99

Department of Mathematics and Computer Science
Bronx Community College

July 6, 2017

Java Workshop Day 2

Java Workshop Day 2

Special Features

Special Features

- The `String` Class is the most common way to represent normal text in a Java program. Other classes, such as `Scanner`, rely on the `String` Class exclusively for text processing.
- The `String` Class is in the `java.lang` package, which is always automatically imported into any Java program.
- It is the only class which overloads the Addition Symbol (+), to represent concatenation. (The Java language does not support operator overloading—this is the only exception.)

A String Object is Composed of chars

A method that returns the chars in a String

`CharAt(int n)` method returns the character at position `n` of a `String`:
If `String s = "abcde";` then `s.CharAt(3)` returns char `'d'`.

Java Workshop Day 2

Simple Decisions—if Statements

Action Only Needed If Condition is true

```
import java.util.Scanner;
public class LetterGrade
{
   public static void main(String[] args)
   {
      Scanner input = new Scanner(System.in);
      System.out.print("Enter Your Average:  ");
      int average = input.nextInt();
      if (average >= 60.0)
         System.out.printLn("You Passed!");
   }
}
```

Two-Way Decisions—if-else Statements

Action Whether Condition is true or false

```
   public static void main(String[] args)
   {
      Scanner input = new Scanner(System.in);
      System.out.print("Enter Your Average:  ");
      int average = input.nextInt();
      String letterGrade = "";
      if (average >= 60.0)
         letterGrade = "P";
      else
         letterGrade = "F";
      System.out.printf("Your Grade is %s",
letterGrade);
   }
```

Two-Way Decisions—condition?a:b Operator

Value Depends on Condition (if true a, if false b)

```
public static void main(String[] args)
{
   Scanner input = new Scanner(System.in);
   System.out.print("Enter Your Average:  ");
   int average = input.nextInt();
   System.out.printf("Your Grade is %s",
         average >= 60?"P":"F");
}
```

Multi-Way Decisions—Nested `if-else`

Many Conditions, Each Has an Action

```
    if (average >= 90.0)
      letterGrade = "A";
    else if (average >= 80.0)
      letterGrade = "B";
    else if (average >= 70.0)
      letterGrade = "C";
    else if (average >= 60.0)
      letterGrade = "D";
    else
      letterGrade = "F";
    System.out.printf("Your Grade is %s",
letterGrade);
```

Multi-Way Decisions—switch Statements

Each Condition is a Value of a Variable

```
boolean passed;
switch(letterGrade)
{
   case('A'):
   case('B'):
   case('C'):
   case('D'):
      passed = true;
   case('F'):
      passed = false;
}
```

Java Workshop Day 2

Infinite Loops

Not Intended to Terminate

```
while (true)
{
  // code for event handling, say
  // like serving web page requests
}
```

Indefinite Loops

Iterations Terminate but Number is not Fixed

```
   // Add an arbitrary number of input positive ints
   int n = 0, total = 0;
   while (n != -1) //Sentinel value -1 is not a valid
input
   {
      total += n;
      n = input.nextInt(); // enter -1 to exit loop
   }
   System.out.printf("Sum is %d", total);
```

Definite (Counter-Controlled) while Loops

Counter managed in different parts of code

```
int total = 0;
int n = 1; // initialize before loop
// find the sum of the integers from 1 to 10
while (n <= 10) // test loop condition
{
   total += n;
   n++; // increment at end of each iteration
}
System.out.printf("Sum is %d", total);
```

do-while Loop Tests Condition After Each Iteration, not Before

At Least One Iteration—The First—Must Happen

```
  // Add an arbitrary number of input positive ints
  int n = 0, total = 0;
  do
  {
    total += n;
    n = input.nextInt(); // enter -1 to exit loop
  }
  while (n != -1) //Sentinel value -1 is not a valid
input
  System.out.printf("Sum is %d", total);
```

Counter is managed in for statement

Initialize, test loop condition, increment

```
int total = 0;
// find the sum of the integers from 1 to 10
for (n = 1; n <= 10; n++)
   total += n;
System.out.printf("Sum is %d", total);
```

break Statements in a Loop

Exits loop

```
int total = 0;
// find the sum of the integers from 1 to 10
for (n = 1; n <= 10; n++)
{
   if (n == 5)
      break;
   total += n;
}
System.out.printf("Sum is %d", total);
```

break Statements in a switch

Keeps from Falling Through to Next case

```
boolean passed;
switch(letterGrade)
{
   case('F'):
      passed = false;
      break;
   case('A'):
   case('B'):
   case('C'):
   case('D'):
      passed = true;
}
```

continue Statements

Exits current iteration, but loop continues

```
int total = 0;
// find the sum of the integers from 1 to 10
for (n = 1; n <= 10; n++)
{
   if (n == 5)
      continue;
   total += n;
}
System.out.printf("Sum is %d", total);
```