

Abstract, Static and Final

2 – 4 pm Tuesday 7/22/2008 @JD2211

1

Announcement

- The update syllabus is available.

2

Agenda

- Review
 - NetBeans and Inheritance
- Abstract class and abstract method
- Overriding and overloading
- Static method and static variable
- Final class and final method

3

Inheritance

- Inheritance is the process of deriving a new class from an exiting one.
- One purpose of inheritance is to reuse existing software
- Inheritance is the *is-a* relationship.

4

NetBeans

- A free, open-source Integrated Development Environment (IDE) for software developers.
- <http://www.netbeans.org/>
- Features
 - <http://www.netbeans.org/features/index.htm>

5

Abstract Class

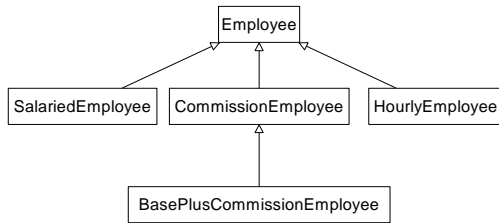
- An abstract class represents a **generic concept** in a class hierarchy (**abstract superclass**)
 - It represents a concept on with other classes can build their definitions

6

Abstract Class

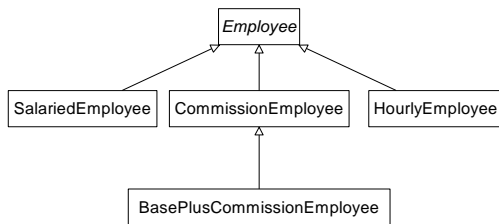
- An abstract class **cannot be instantiated** and usually contains **one or more abstract methods**, which has **no definition**
- **Concrete class**

Payroll System – UML Class Diagram I



8

Payroll System – UML Class Diagram II



9

Abstract Class Employee

- Class Employee is a "generic" employee
- The classes that extend Employee are
 - SalariedEmployee
 - HourlyEmployee
 - CommissionEmployee
 - BasePlusCommissionEmployee

10

Quiz

- Among the methods used in the 7-22 lab, which method(s) can be generic?

11

Abstract Class in Java

- A class is made abstract by declaring it with keyword abstract
 - public **abstract** class Employee {
- An abstract class normally contains one or more abstract methods
 - public **abstract** double earnings();

12

Abstract Method

- Abstract methods do **not** provide implementations
- A class that contains abstract methods **must** be declared as an abstract class

13

Overriding Methods

- When a child class defines a method with the same name and signature as a method in the parent class, we say that the child's version **overrides** the parent's version in favor of its own
- An abstract methods is overridden by subclasses

14

Overloading Methods

- In Java, you can use the same method name with different parameter lists for multiple methods
 - This technique is called **method overloading**
- Method overloading is useful when you need to perform similar methods on the different types of data or different number of parameters

15

Overloading Methods - Example

```
public int sum (int a, int b){  
    return a + b;  
}  
  
public int sum (int a, int b, int c){  
    return a + b + c;  
}  
  
println (String s)  
println (int i)  
println (double d)  
println (char c)  
println (boolean b)
```

16

Overloading Constructors

- Constructor can be overloaded, and often are.
- By providing multiple versions of a constructor, we provide multiple ways to initialize an object

17

Static Class Members

- Static (class) methods
- Static (class) variables

18

Quiz

- We've used a static method in previous examples and lab assignments. What is that?

19

static Method

- A class (static) method is one that is not particular to one object but related to the class itself.
- Static methods can be invoked through the class name.
- We do not have to instantiate an object the class in order to invoke the methods.
- All methods in the `Math` class are static methods.

20

static Variables

- So far, we've seen two categories of variable
 - local variables declared in a class
 - local variables declared inside a methods
- A static (class) variable is shared among all instance (objects) of a class
 - There is only one copy of a static variable for all objects of the class

21

final

- final variables
 - Constant
- final methods
 - Cannot be overridden
- final classes
 - Cannot be extended

22

Lab Assignment

- <http://www.csun.edu/~twang/Java/Lab/7-24-Lab.pdf>

23
