Advanced Class Design

Oracle® Certified Professional, Java® SE 7 Programmer

Module 2

Module 2 - Objectives

- Identify when and how to apply abstract classes.
- Construct abstract Java classes and subclasses.
- Use the static and final keywords.
- Create top-level and nested classes.
- Use enumerated types.

Abstract Classes

- A class declared as abstract can not be instantiated, and must be sub-classed.
- A class is must be declared abstract if any method is declared as abstract.
- Fields declared in an abstract class are not public, static, or final by default like an interface.§ 9.3
- If an abstract class contains only abstract methods, it should be declared as an interface.

Abstract Classes (cont.)

- If a class implements an interface, but does not provide an implementation for all of the interface methods, it must be declared as abstract.
- An enum can not be declared as abstract.
- An abstract class is used to encapsulate fields and behaviors which are expected in sub-classes.

final modifier

- final can be applied to classes, fields, and methods.§8.1.1.2, §8.3.1.2, and §8.4.3.3
- A final class can not be sub-classed.
- All methods of a final class are final, and can not be overridden.
- A static final field that is not definitely assigned must be initialized by a static initializer. §8.3.1.2
- A final field that is not definitely assigned must be initialized at the end every constructor. §8.3.1.2

final modifier

- A method can be declared final to prevent sub-classes from overriding, or hiding it.§ 8.4.3.3
- At run-time, a machine-code generator or optimizer can "inline" the body of a final method, replacing an invocation of the method with the code in its body. The inlining process must preserve the semantics of the method invocation. §8.4.3.3
- Local field declarations can be final.
- A resource of a try-with-resources statement (§14.20.3) and an exception parameter of a multi-catch clause (§14.20) are implicitly declared final. §4.12.4

static modifier

- A static class modifier can be only applied to member classes (inner classes). § 8.1.1
- A static field is incarnated when the class in initialized (created). §8.3.1.1
- A static method is also called a class method.
- A static method can be invoked without reference to a specific instance of the object.

static final modifiers

- A field declared as static and final is considered a constant.§9.3
- If the field is a primitive, or String and its value is known at compile time. It is a compile time constant. The compiler replaces all instances with the actual value.

Class Declaration

- A public class must be in a source file which has the same name as the public class name.§7.6
- Any number of individual classes may be in a given source file as long as none are declared public.
- The source file does not need to be named for any of the classes contained.
- import and package declarations apply to all classes within the source file.

Class Declaration (cont.)

- If there is a package declaration, it must be the first item occurring before any imports, or class declarations.
- If there are any imports, they must occur after the package name, and before the class declaration.
- A class may be declared inside of another class (enclosing class). This results in a member class, or inner-class.
- There are two types of nested (inner-classes): static, and instance classes.
- A class may created as an anonymous inner-class by directly implementing an interface.

enum types

- An enum can not be abstract. §8.9
- An enumerated type can not be cloned. §8.9
- It is a compile time error to explicitly instantiate an enum. §15.9.1
- An enum contains constants which form the body of the enumeration. §8.9.1
- You can use == or equals to determine equality since there is only one enum constant. §8.9.1

enum types (cont.)

- It is a compile-time error if a constructor declaration of an enum type is public or protected. §8.9.2
- If an enum type has no constructor declarations, then a private constructor that takes no parameters (to match the implicit empty argument list) is automatically provided. §8.9.
- An enum constant is implicitly static and final. §16.5
- A nested enum is implicitly static. §8.9
- An enum may not be local, nor can it be inside an inner class.

enum types (cont.)

- An enum can be used in place of a primitive int type in a case statement.
- The use of enum in a case statement is a vast improvement over using simple primitives.

enum example

```
public enum Coin {

PENNY(1), NICKEL(5), DIME(10), QUARTER(25);

Coin(int value) { this.value = value; }

private final int value;

public int value() { return value; }

}
```

References

- Java™ Language Specification, Java SE 7 Edition, James Gosling, Bill Joy, Guy Steele, Gilad Bracha, and Alex Buckley.
- Preparation for Java Programmer Language Certification
- Sun® Certified Java™ Programmer for Java™ 6 Study Guide, Kathy Sierra, and Bert Bates

creative commons

Attribution-NonCommercial-ShareAlike 3.0 Unported