STATIC FIELDS AND METHODS

zyBook 9.15

Oracle - Java Tutorial: Understanding Class Members

@ Dr. Jessica Young Schmidt and NCSU Computer Science Faculty

STATIC FIELD

- The keyword static indicates a variable is allocated in memory only once during a program's execution. Static variables reside in the program's static memory region and have a global scope. Thus, static variables can be accessed from anywhere in a program.
- In a class, a static field is a field of the class instead of a field of each class object. Thus, static fields are independent of any class object, and can be accessed without creating a class object. Static fields are declared and initialized in the class definition. Within a class method, a static field is accessed using the field name. A public static field can be accessed outside the class using dot notation: ClassName.fieldName.
- Static fields are also called class variables, and non-static fields are also called instance variables.

STATIC METHOD

- A static member method is a class method that is independent of class objects. Static member methods are typically used to access and mutate private static fields from outside the class. Since static methods are independent of class objects, the this parameter is not passed to a static member method. So, a static member method can only access a class' static fields.

Example from zyBook, Store.java:

```java
public class Store {
   // Declare and initialize private static field
   private static int nextId = 101;

   // Define private fields
   private String name;
   private String type;
   private int id;

   public Store(String storeName,
                String storeType) {
      name = storeName;
      type = storeType;
      id = nextId;

      // Increment each time a Store
      // object is created
      ++nextId;
   }

   public int getId() {
      return id;
   }

   public static int getNextId() {
      return nextId;
   }
}
```

NewStores.java:

```java
public class NewStores {
   public static void main(String[] args) {
      Store store1 = new Store("Macy's",
                               "Department");
      Store store2 = new Store("Albertsons",
                               "Grocery");
      Store store3 = new Store("Ace",
                               "Hardware");

      System.out.println("Store 1's ID: "
                         + store1.getId());
      System.out.println("Store 2's ID: "
                         + store2.getId());
      System.out.println("Store 3's ID: "
                         + store3.getId());
      System.out.println("Next ID: "
                         + Store.getNextId());
   }
}
```

```
$ java -cp bin NewStores
Store 1's ID: 101
Store 2's ID: 102
Store 3's ID: 103
Next ID: 104
```