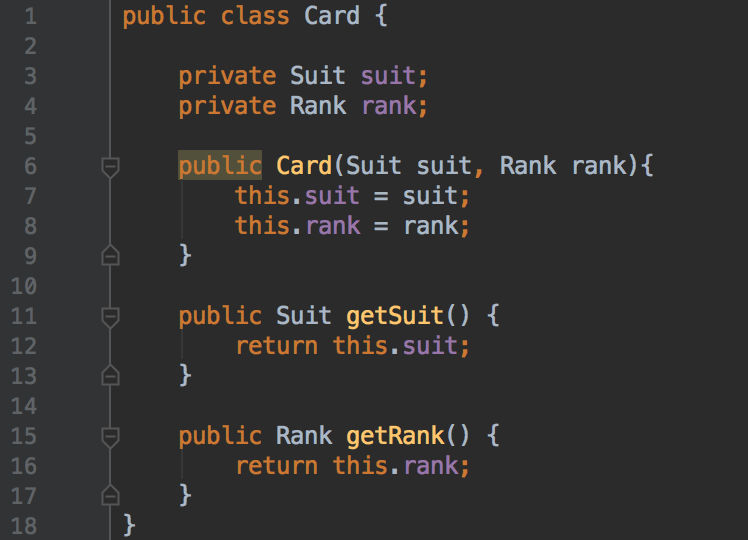
# Implementation and Testing Unit – Evidence

## **David Ellis**

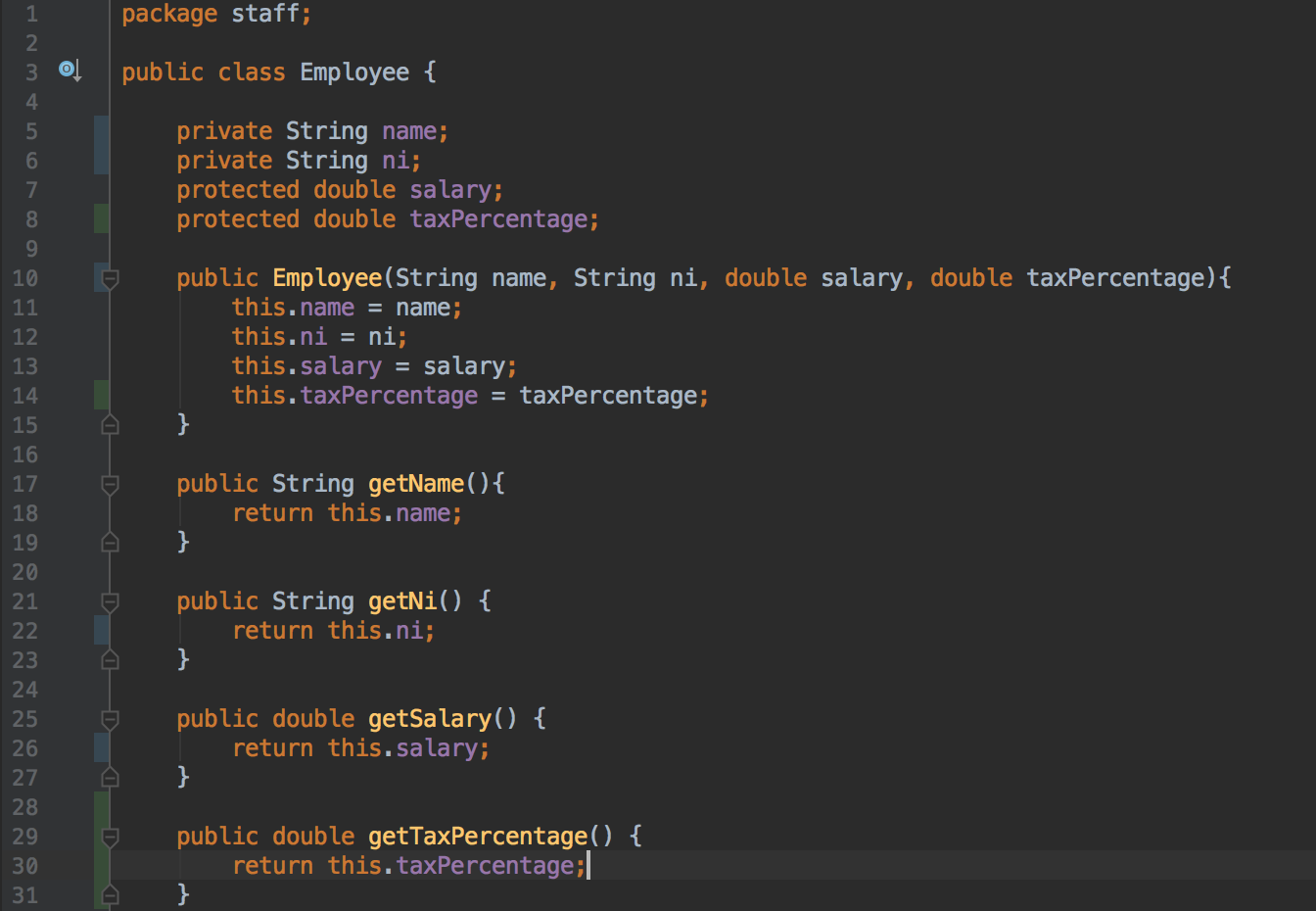
## **Cohort E18**

## I.T. 1 – Encapsulation Example Screenshot



## I.T. 2 – Inheritance Example Screenshots

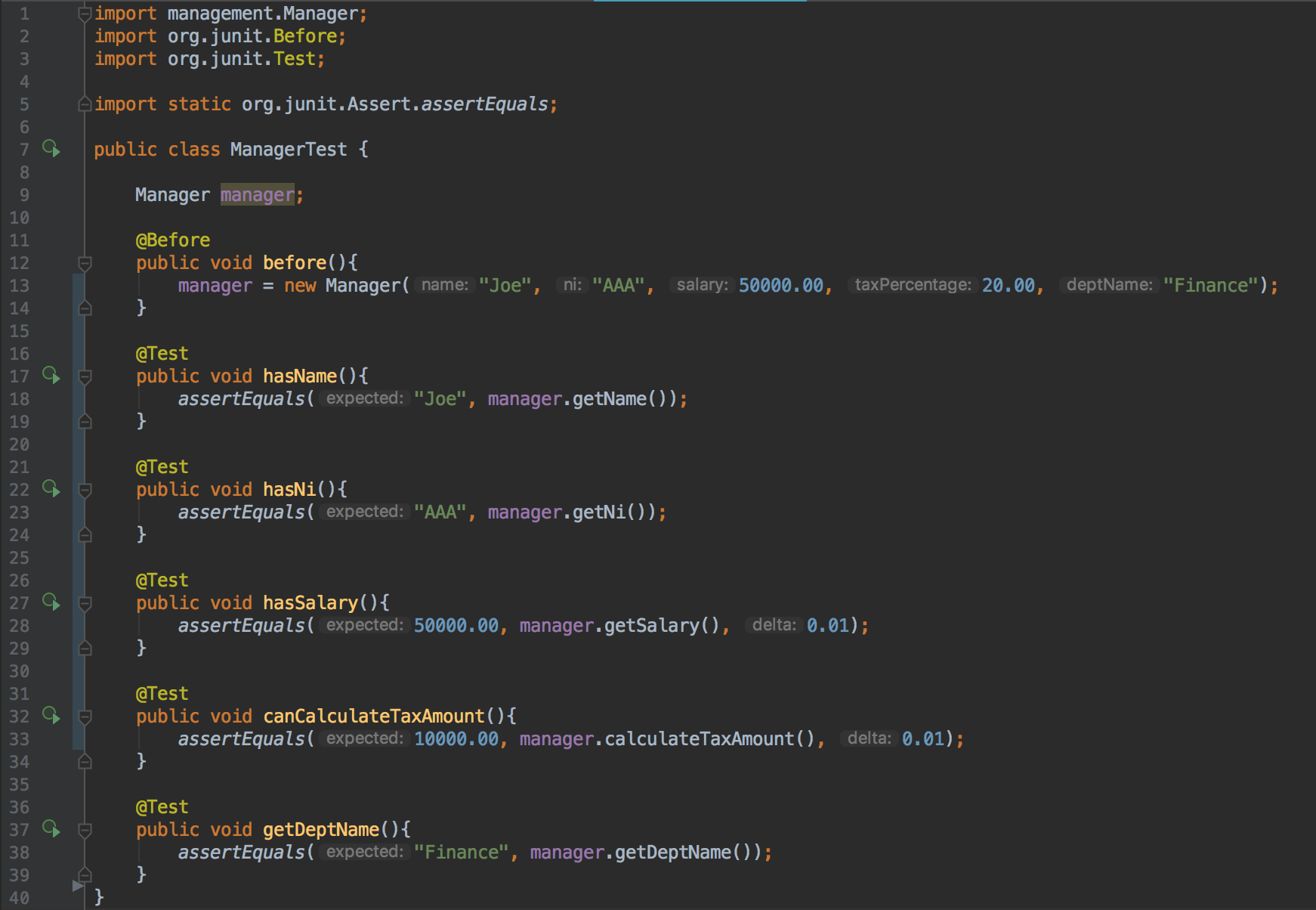
A class:



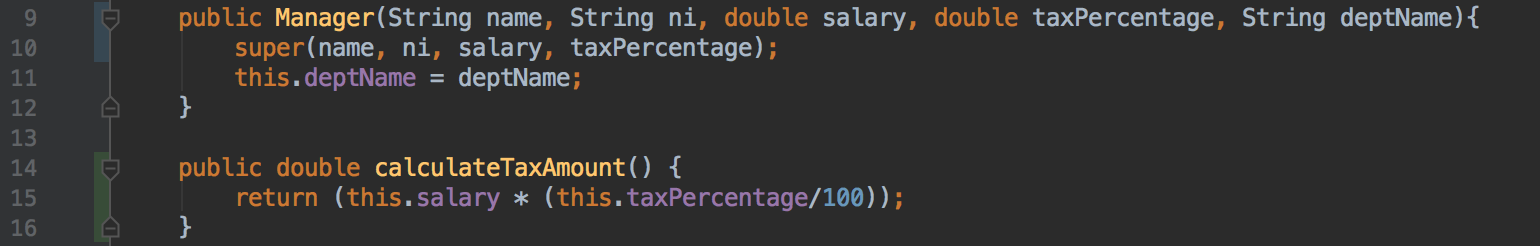
A class that inherits from the previous class:



An object in the inherited class:

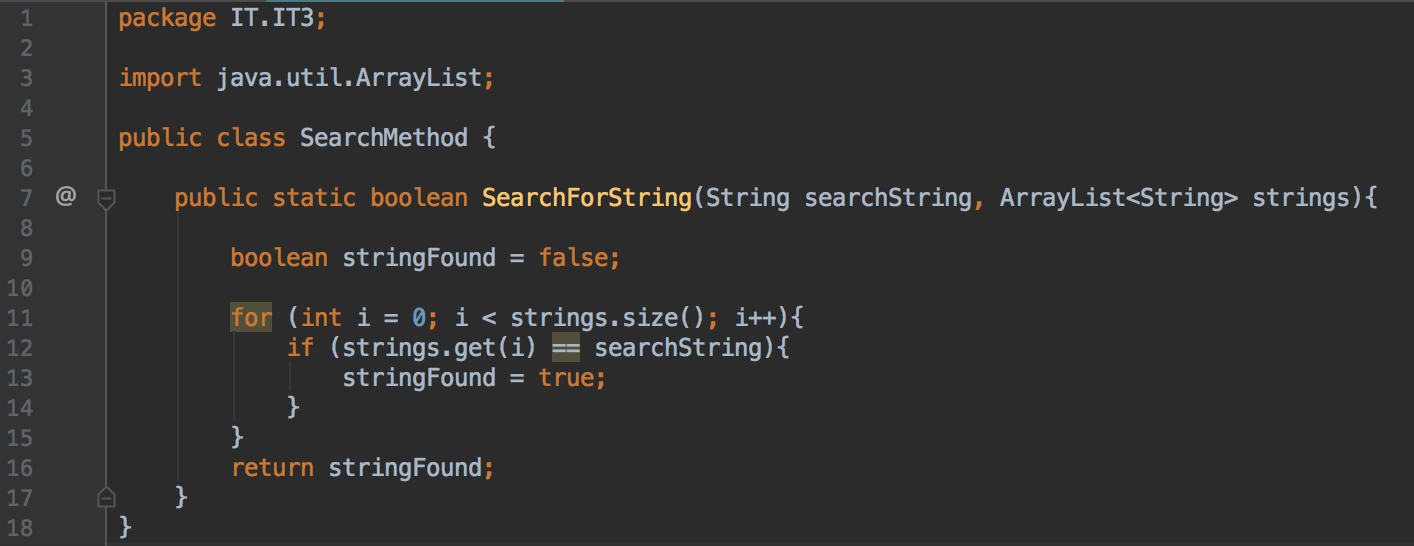


A method that uses the information inherited from another class:

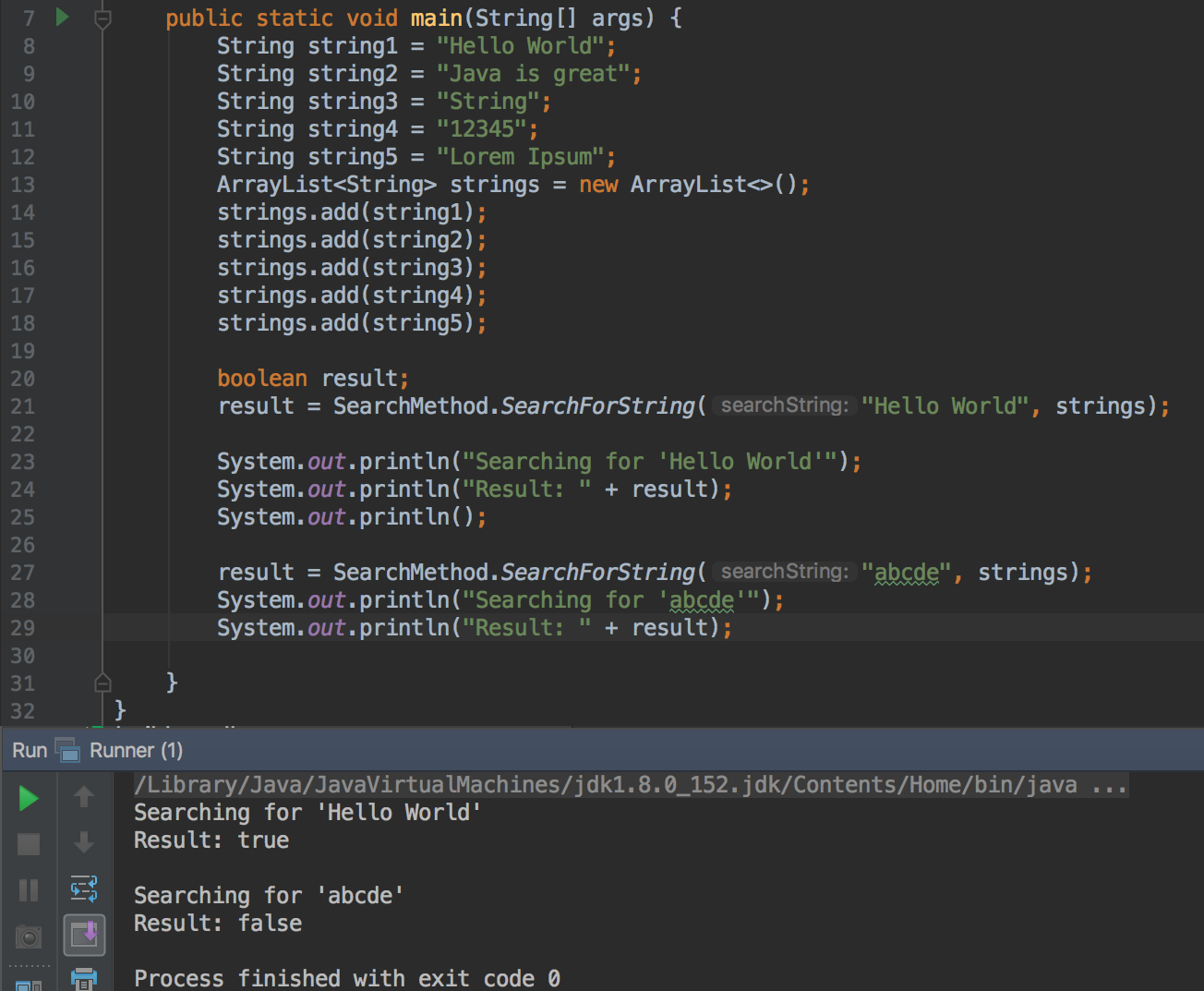


## I.T. 3 – Searching Data Example Screenshots

The searchForString method searches for a specific string in an ArrayList of String objects and returns true if it finds the string or false if it does not find the string.



Result of calling SearchForString method:

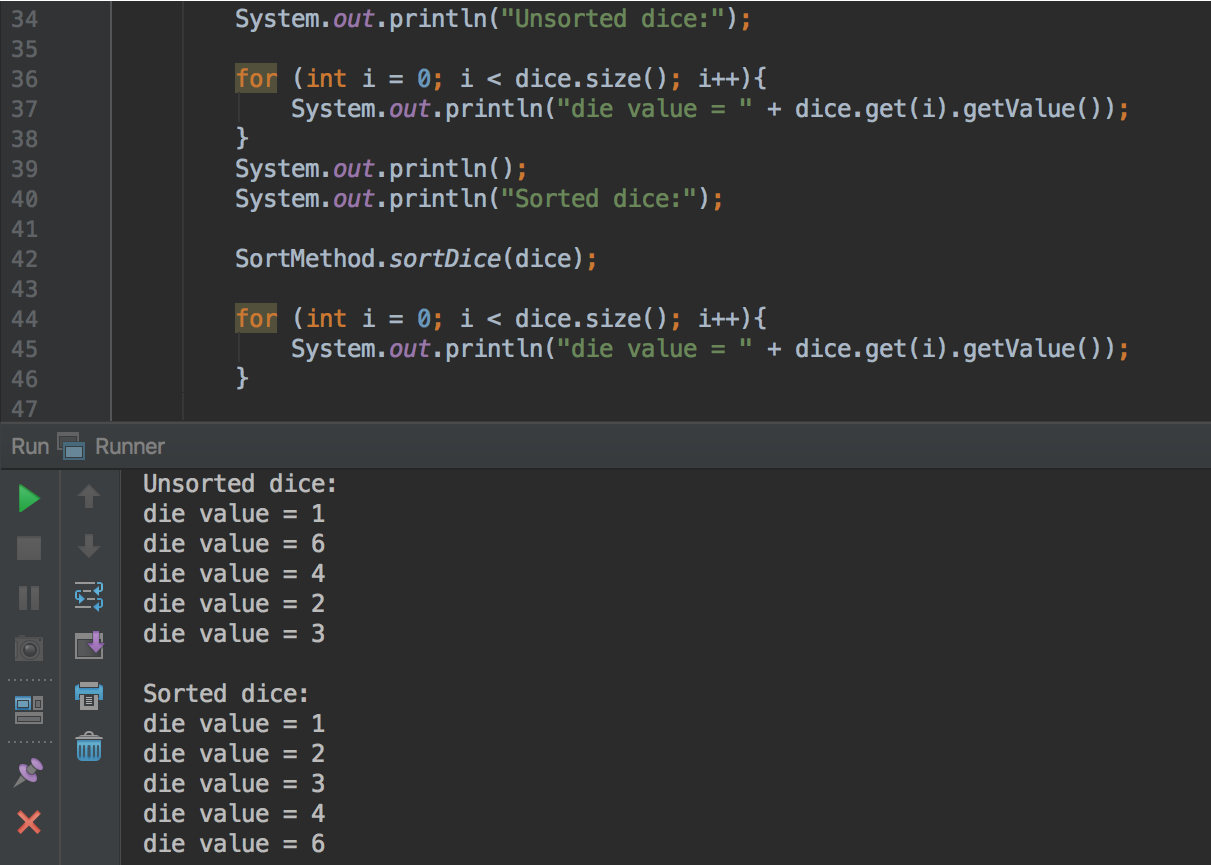


## I.T. 4 – Sorting Data Example Screenshots

The sortDice method sorts an ArrayList of Die objects (a Die object has a value parameter) by using a comparator to tell Collections.sort that it needs to sort the collection of dice based on the die value.

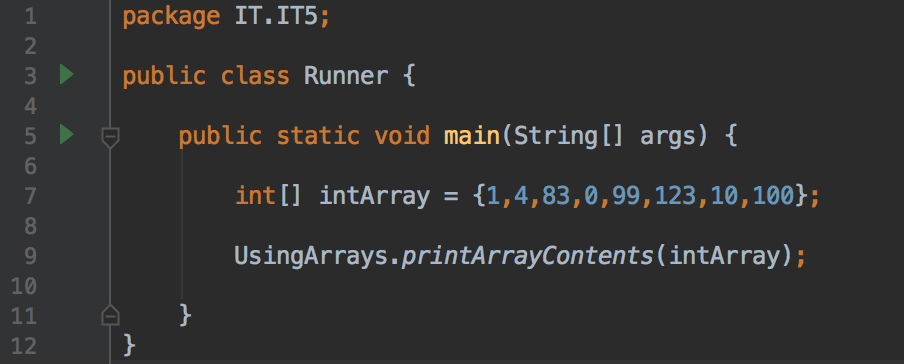


Result of calling the sortDice method:

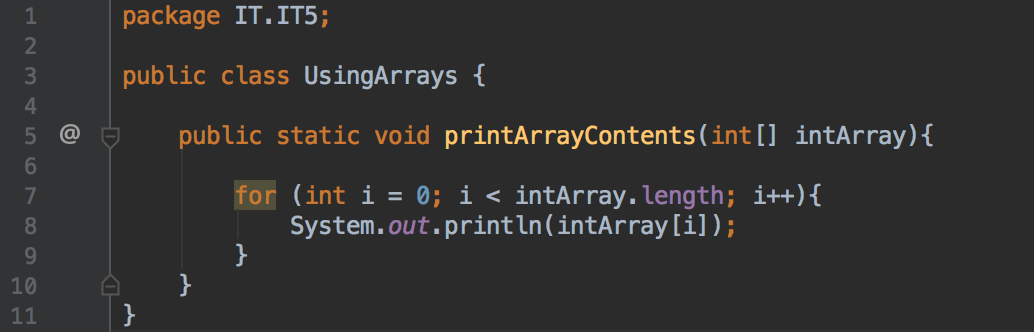


## I.T. 5 – Array Example Screenshots

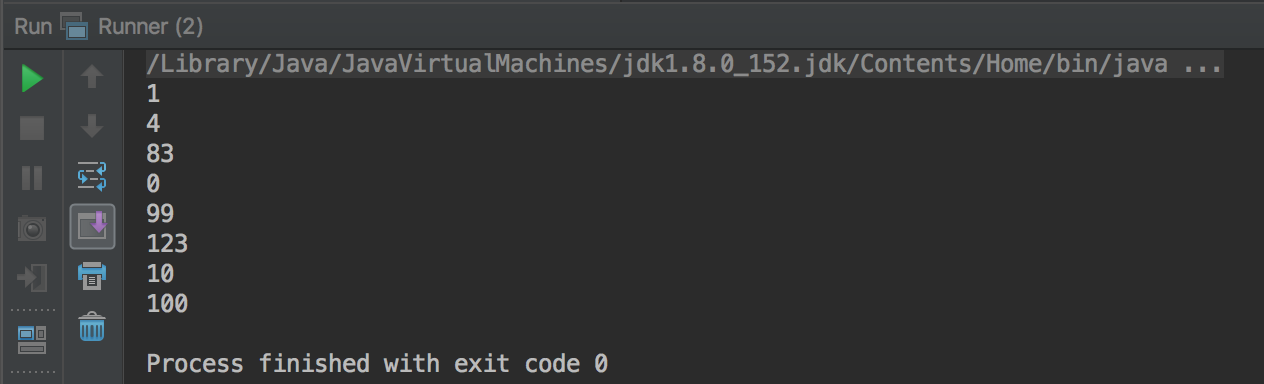
An array of integers containing 8 elements is created in the Runner class below



The printArrayContents method shown below takes in an array of integers and prints out the contents of the array



The output from running the Runner class is shown below, with the contents of the array of integers printed out

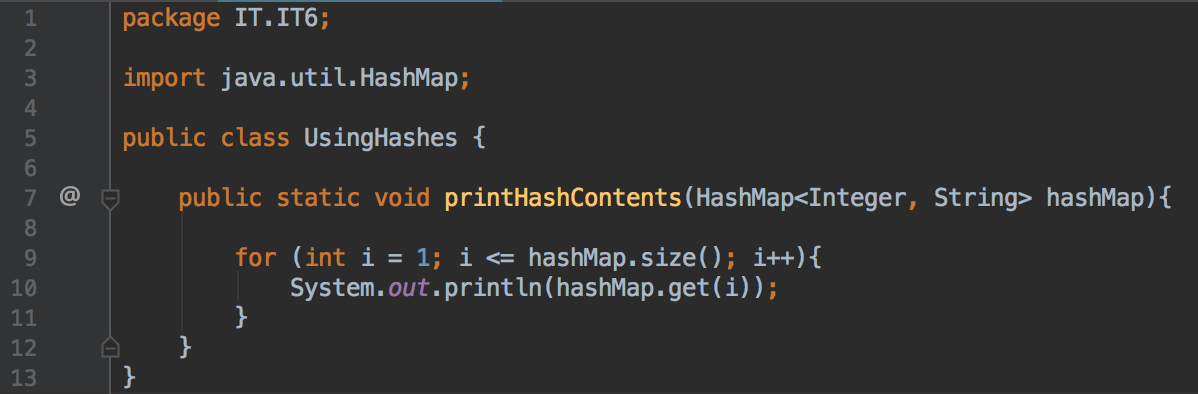


## I.T. 6 – Hash Example Screenshots

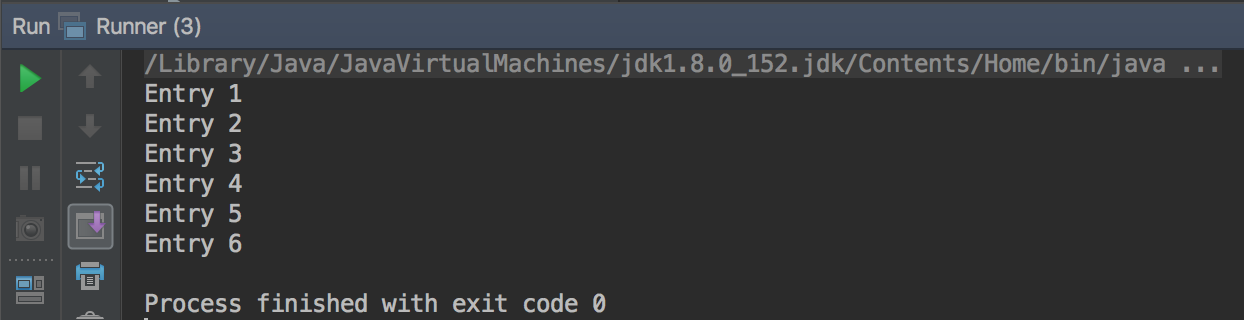
A hashmap of integers and strings containing 6 entries is created in the Runner class below



The printHashContents method shown below takes in a hashmap of integers and strings and prints out the contents of the hashmap



The output from running the Runner class is shown below, with the string values of the hashmap printed out

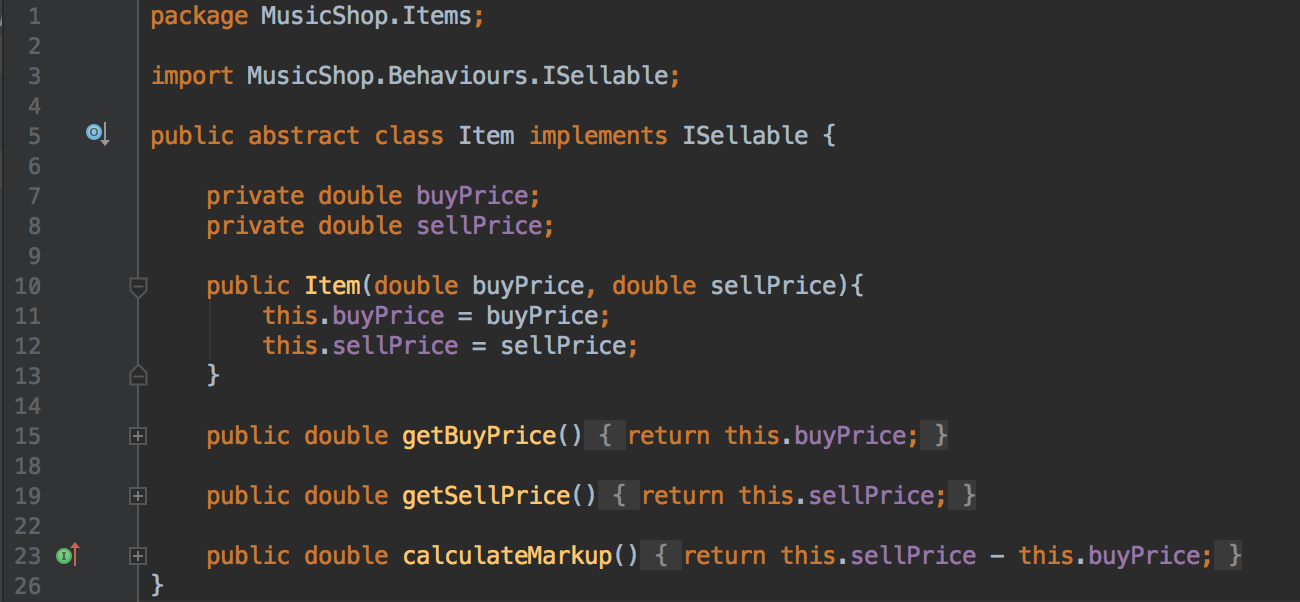


## I.T. 7 – Polymorphism Example Screenshots

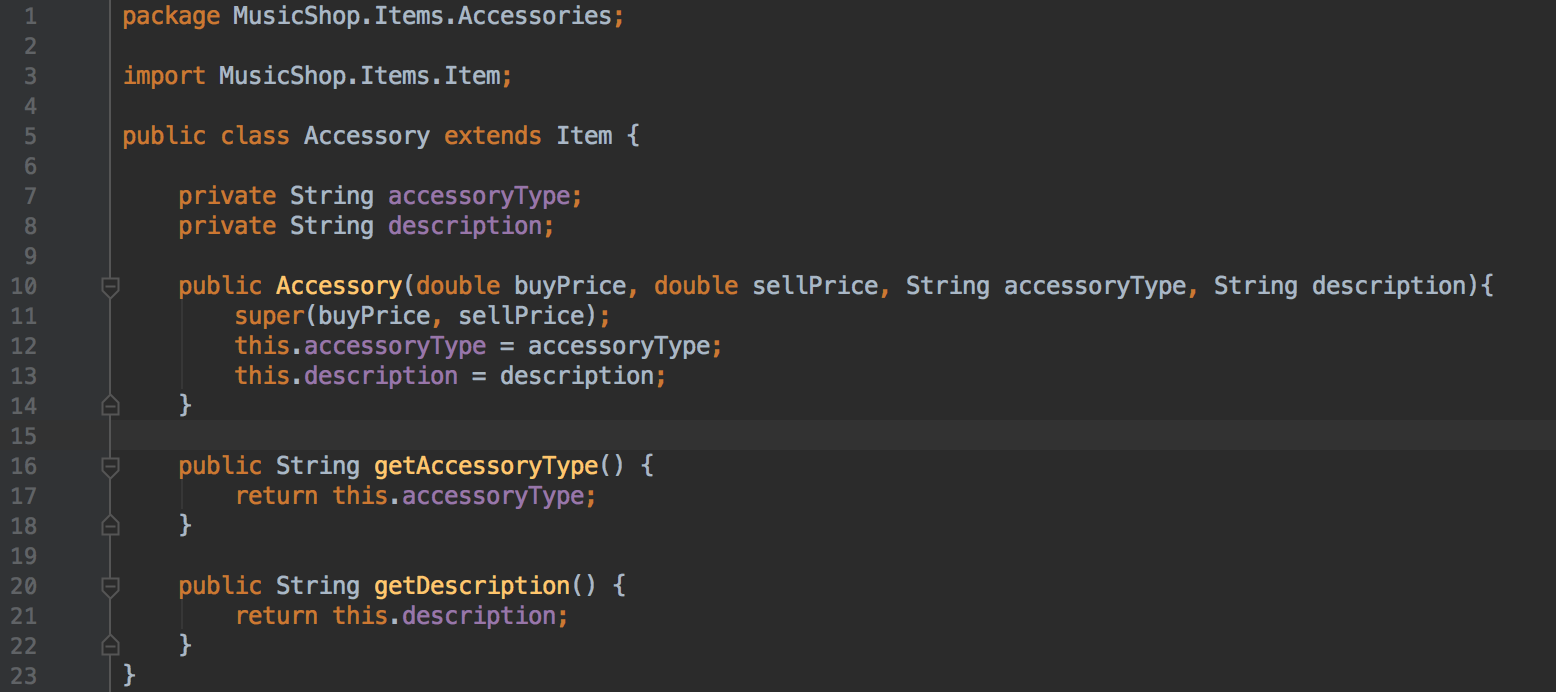
**MusicShop** class which has an ArrayList of ISellable item objects:



**Item** class, which implements the **ISellable** interface:



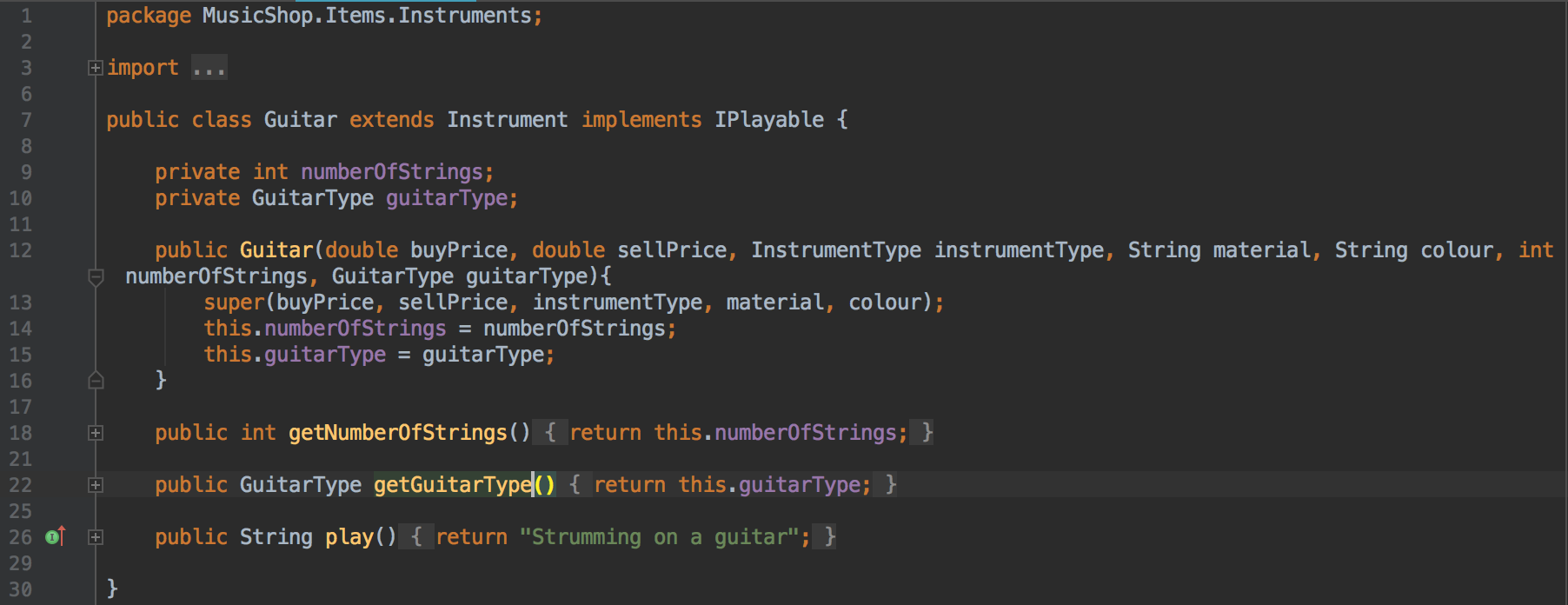
**Accessory** class, which extends the **Item** class (and therefore implements the **ISellable** interface):



**Instrument** class, which extends the **Item** class (and therefore implements the **ISellable** interface):



**Guitar** class, which extends the **Instrument** class (and therefore implements the **ISellable** interface):



**ISellable** interface:

