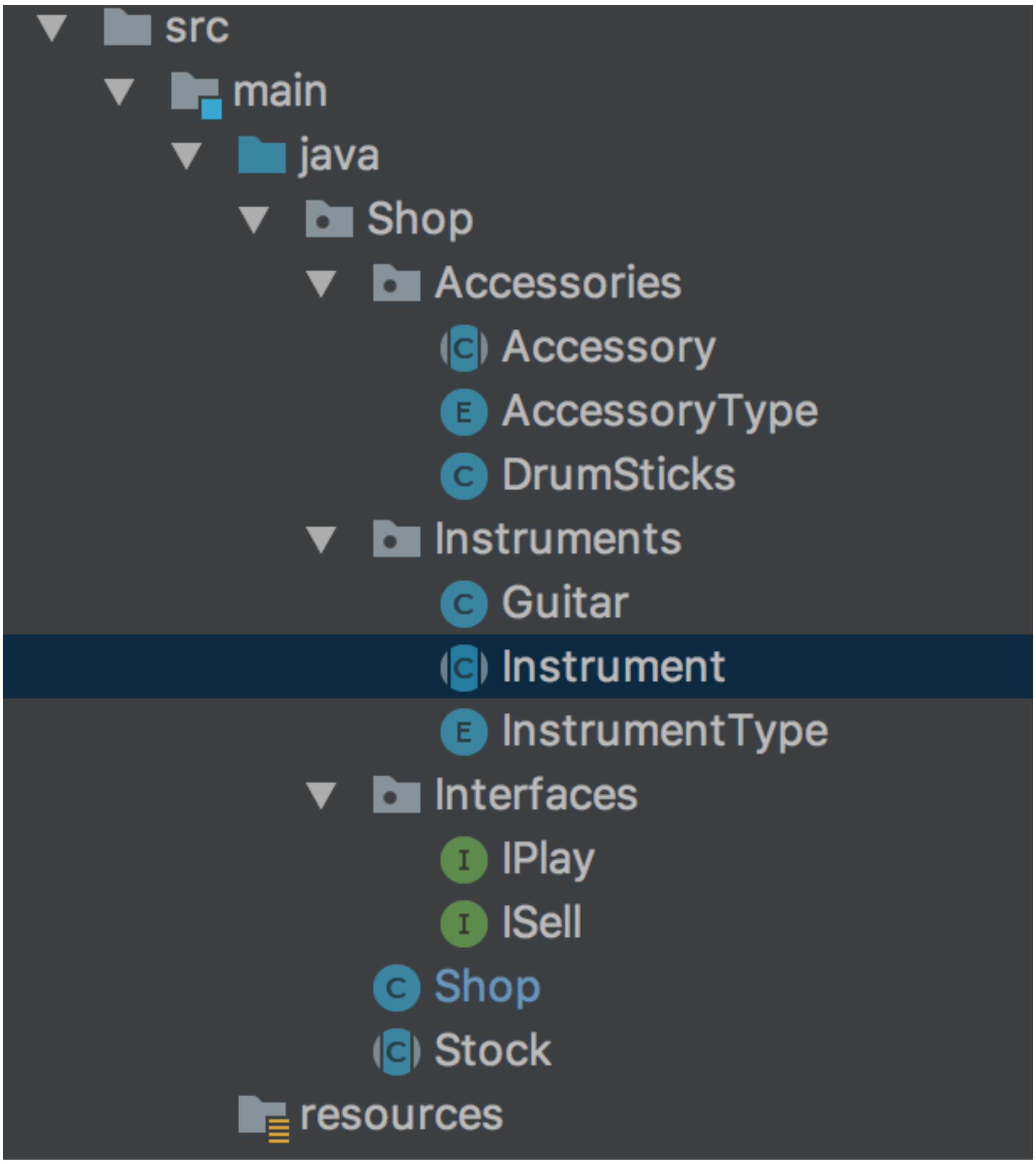
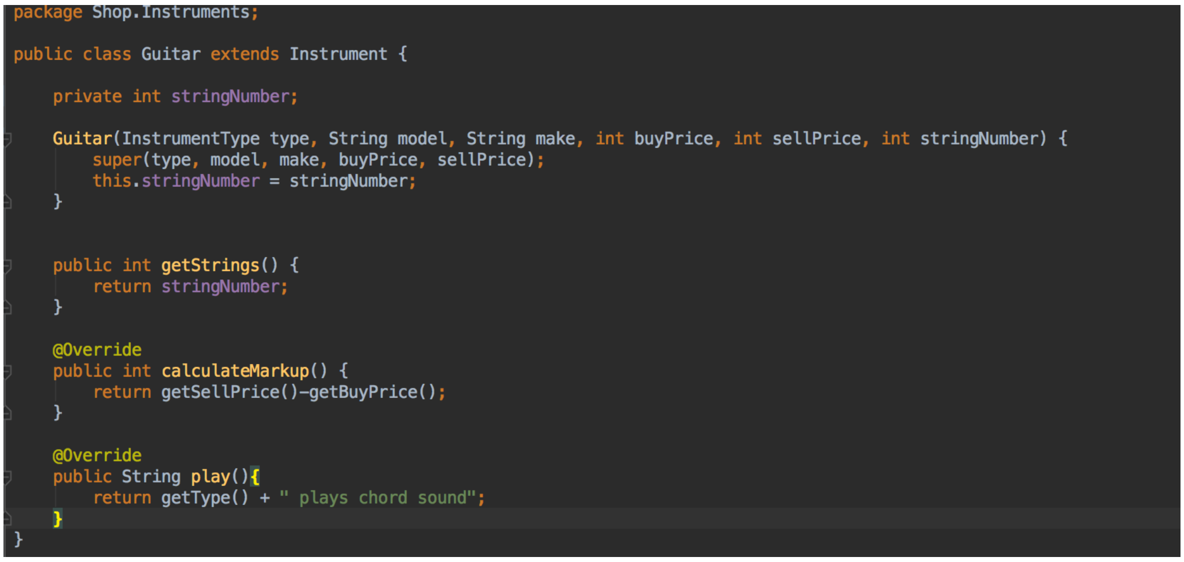
Evidence for Implementation and Testing Unit

Name: Jussi Pardoe
Cohort: e19

I.T 1 Example of Encapsulation

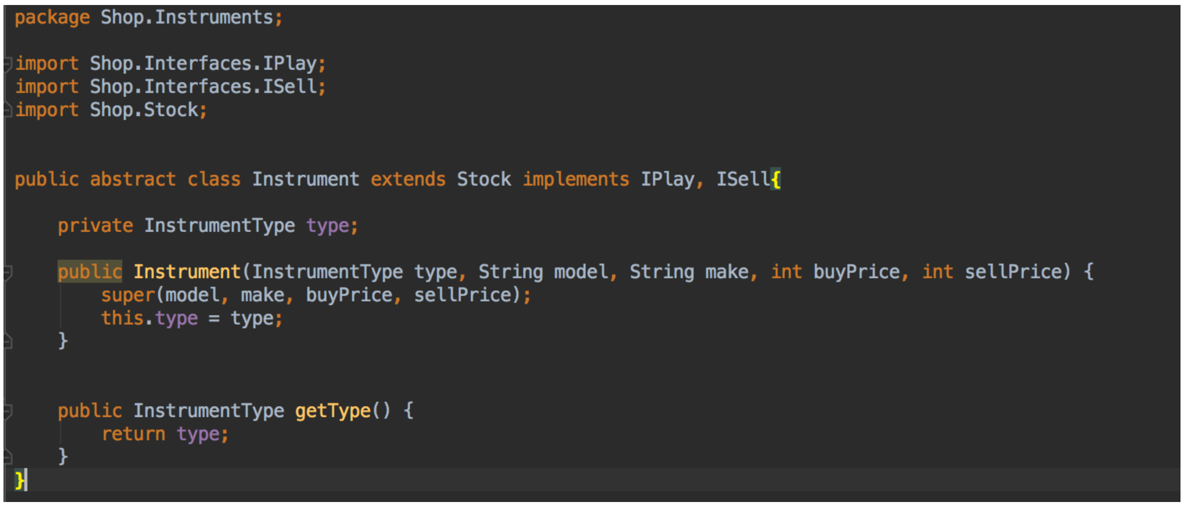




```java
package Shop.Instruments;

public class Guitar extends Instrument {

    private int stringNumber;

    Guitar(InstrumentType type, String model, String make, int buyPrice, int sellPrice, int stringNumber) {
        super(type, model, make, buyPrice, sellPrice);
        this.stringNumber = stringNumber;
    }


    public int getStrings() {
        return stringNumber;
    }

    @Override
    public int calculateMarkup() {
        return getSellPrice()-getBuyPrice();
    }

    @Override
    public String play(){
        return getType() + " plays chord sound";
    }
}
```

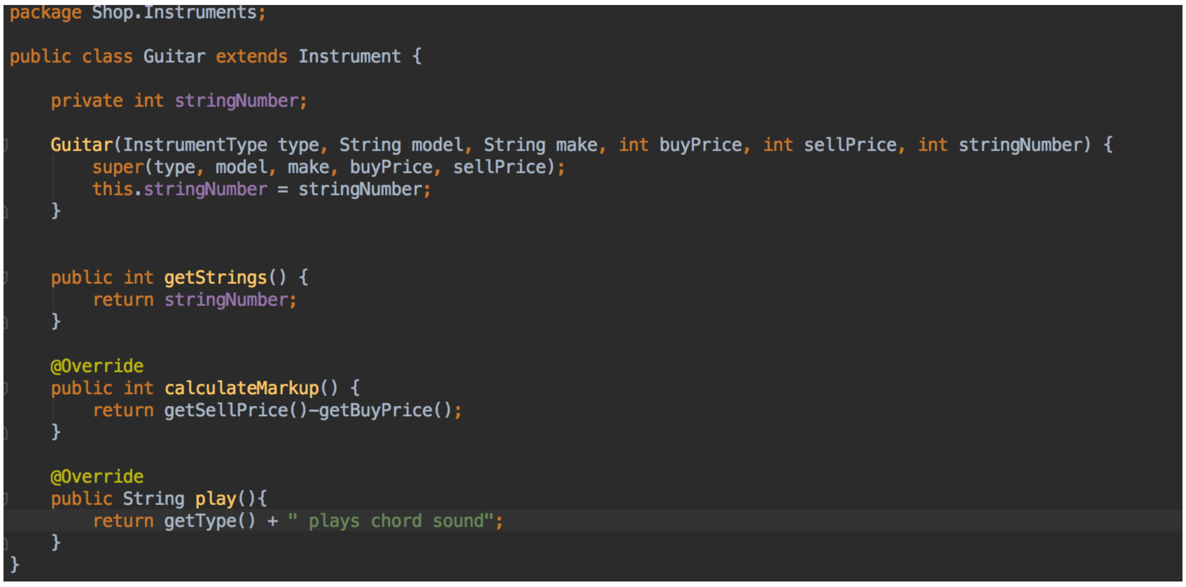
I.T 2 Example of Inheritance

class -



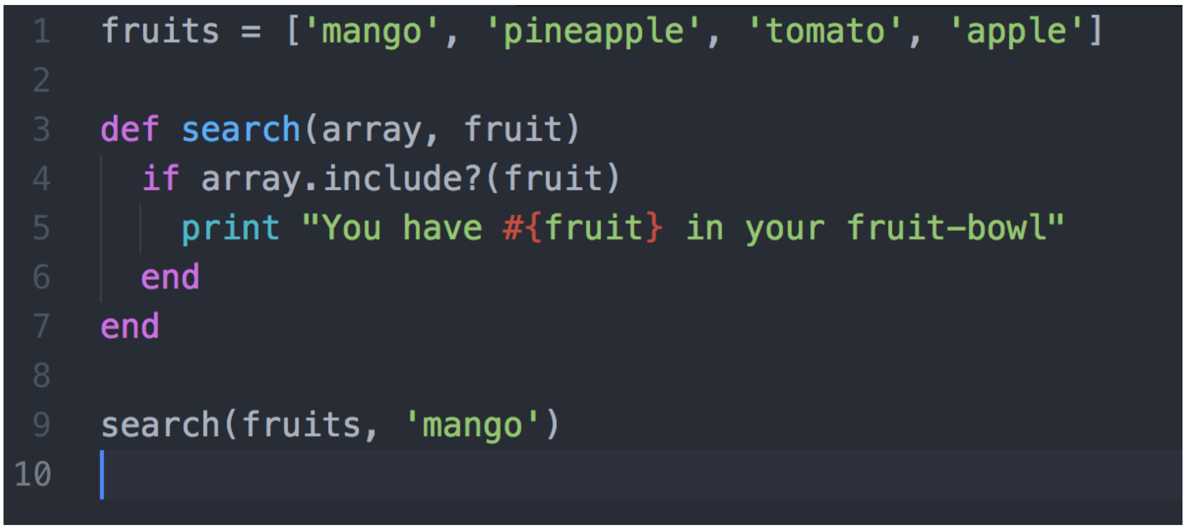
```java
package Shop.Instruments;

import Shop.Interfaces.IPlay;
import Shop.Interfaces.ISell;
import Shop.Stock;


public abstract class Instrument extends Stock implements IPlay, ISell{

    private InstrumentType type;

    public Instrument(InstrumentType type, String model, String make, int buyPrice, int sellPrice) {
        super(model, make, buyPrice, sellPrice);
        this.type = type;
    }


    public InstrumentType getType() {
        return type;
    }
}
```

class that inherits -



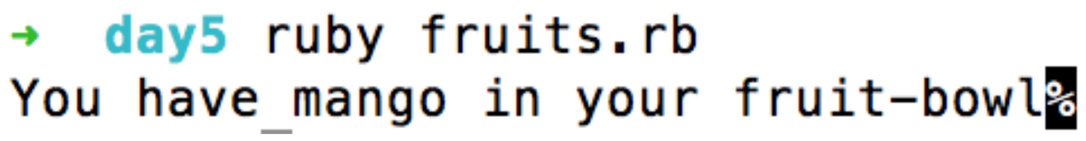
```java
package Shop.Instruments;

public class Guitar extends Instrument {

    private int stringNumber;

    Guitar(InstrumentType type, String model, String make, int buyPrice, int sellPrice, int stringNumber) {
        super(type, model, make, buyPrice, sellPrice);
        this.stringNumber = stringNumber;
    }


    public int getStrings() {
        return stringNumber;
    }

    @Override
    public int calculateMarkup() {
        return getSellPrice()-getBuyPrice();
    }

    @Override
    public String play(){
        return getType() + " plays chord sound";
    }
}
```

I.T 3 Example of Searching
code:



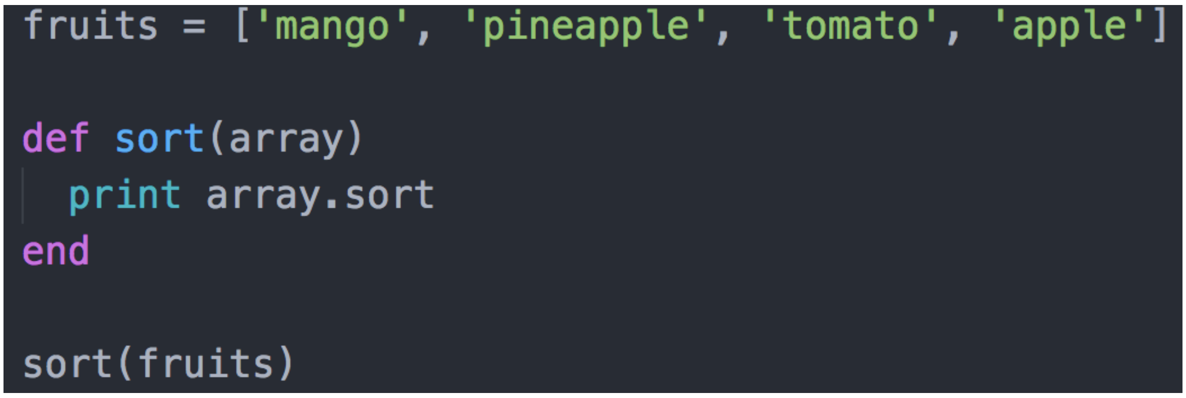
```ruby
fruits = ['mango', 'pineapple', 'tomato', 'apple']

def search(array, fruit)
  if array.include?(fruit)
    print "You have #{fruit} in your fruit-bowl"
  end
end

search(fruits, 'mango')

```

result:

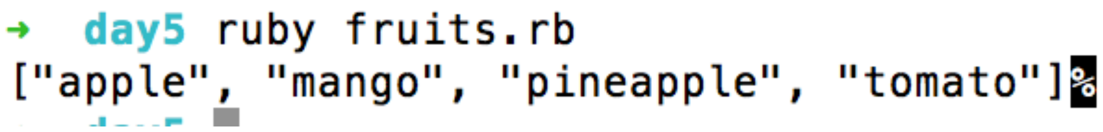


```
➜  day5 ruby fruits.rb
You have mango in your fruit-bowl%
```

I.T 4 Example of Sorting

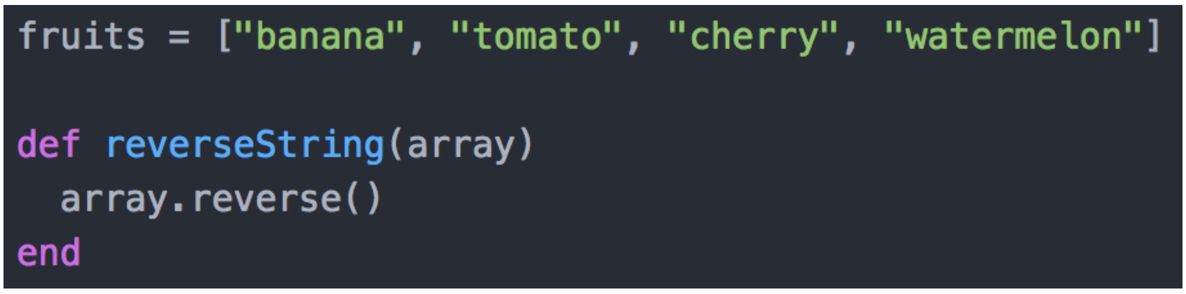


```ruby
fruits = ['mango', 'pineapple', 'tomato', 'apple']

def sort(array)
  print array.sort
end

sort(fruits)
```

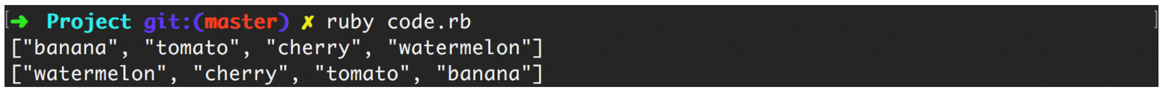


```
➜  day5 ruby fruits.rb
["apple", "mango", "pineapple", "tomato"]%
```

I.T 5 Example of an Array, a function that uses an array and the result

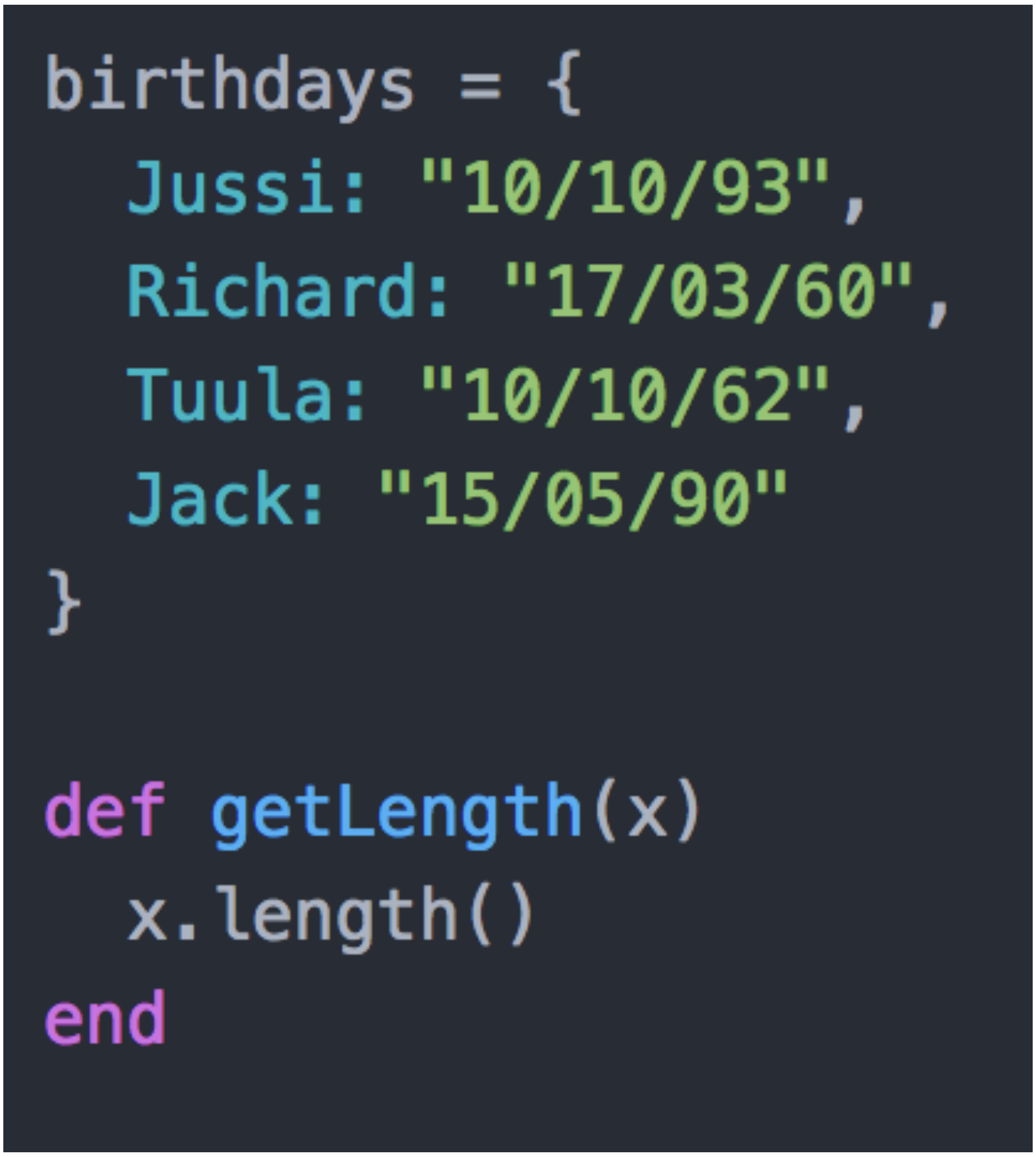


```ruby
fruits = ["banana", "tomato", "cherry", "watermelon"]

def reverseString(array)
  array.reverse()
end
```

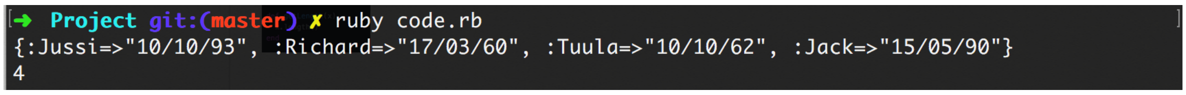


```
→ Project git:(master) ✗ ruby code.rb
["banana", "tomato", "cherry", "watermelon"]
["watermelon", "cherry", "tomato", "banana"]
```

I.T 6 Example of a Hash, a function that uses a Hash and the result

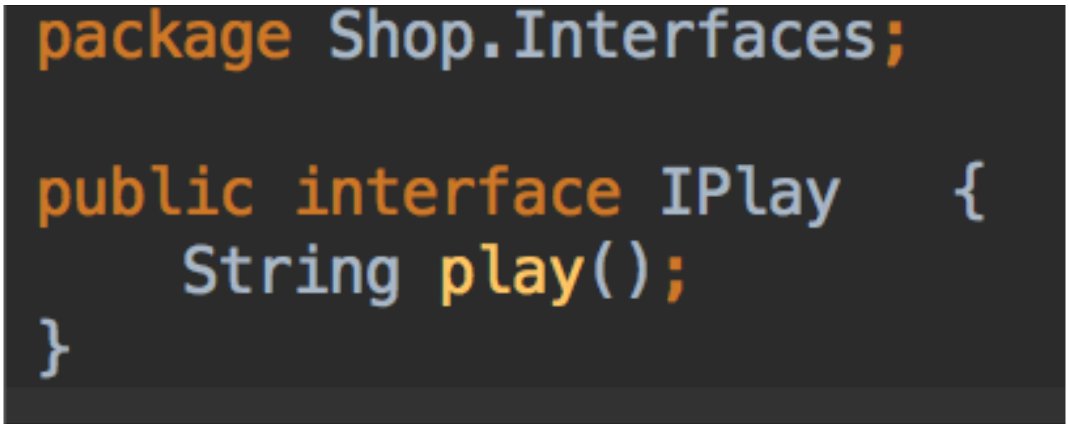


```ruby
birthdays = {
  Jussi: "10/10/93",
  Richard: "17/03/60",
  Tuula: "10/10/62",
  Jack: "15/05/90"
}

def getLength(x)
  x.length()
end
```

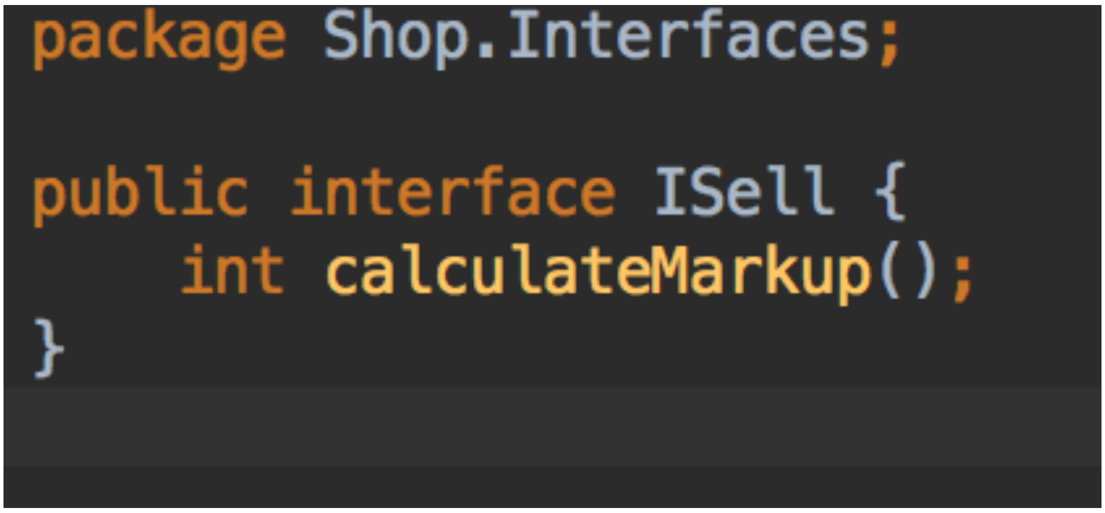


```
→ Project git:(master) ✗ ruby code.rb
{:Jussi=>"10/10/93", :Richard=>"17/03/60", :Tuula=>"10/10/62", :Jack=>"15/05/90"}
4
```

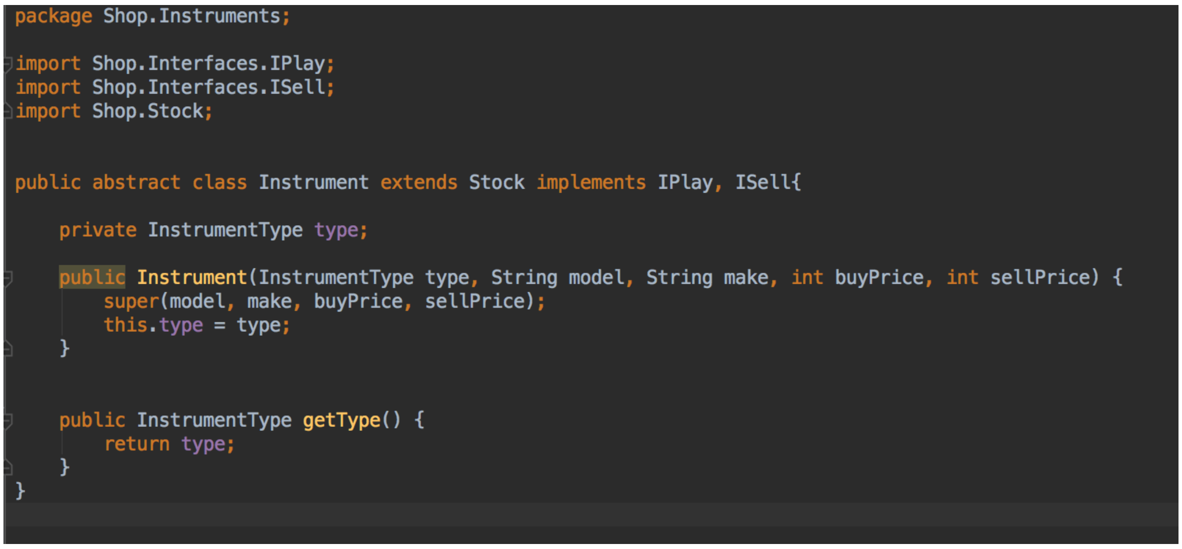
I.T 7 Example of polymorphism in a program



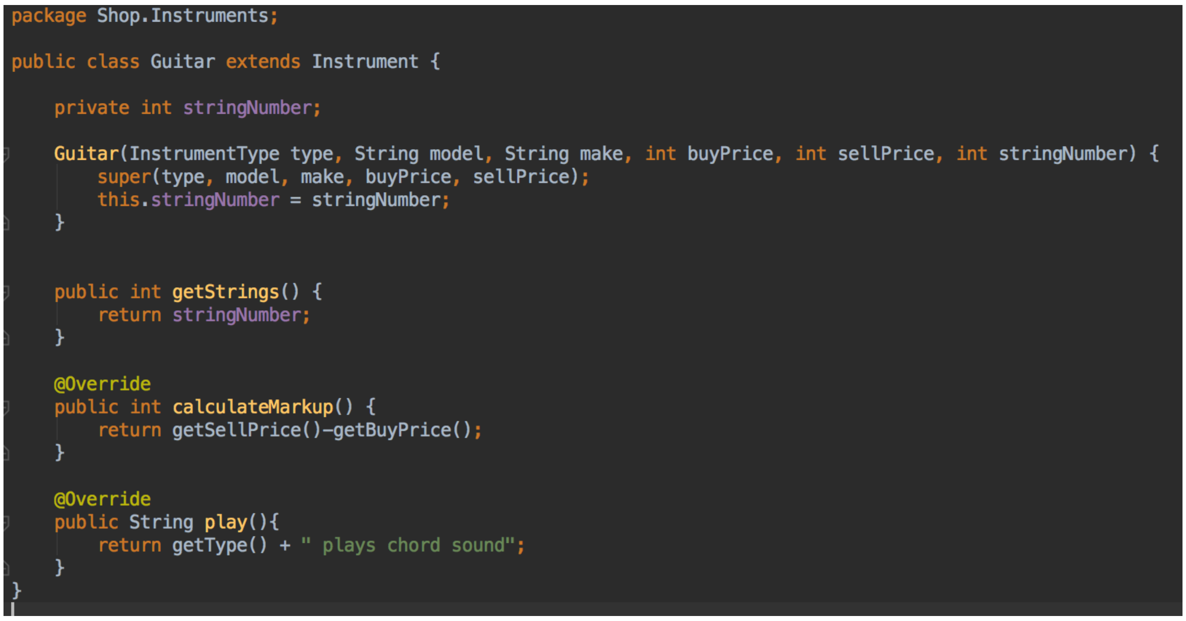
```java
package Shop.Interfaces;

public interface IPlay   {
    String play();
}
```



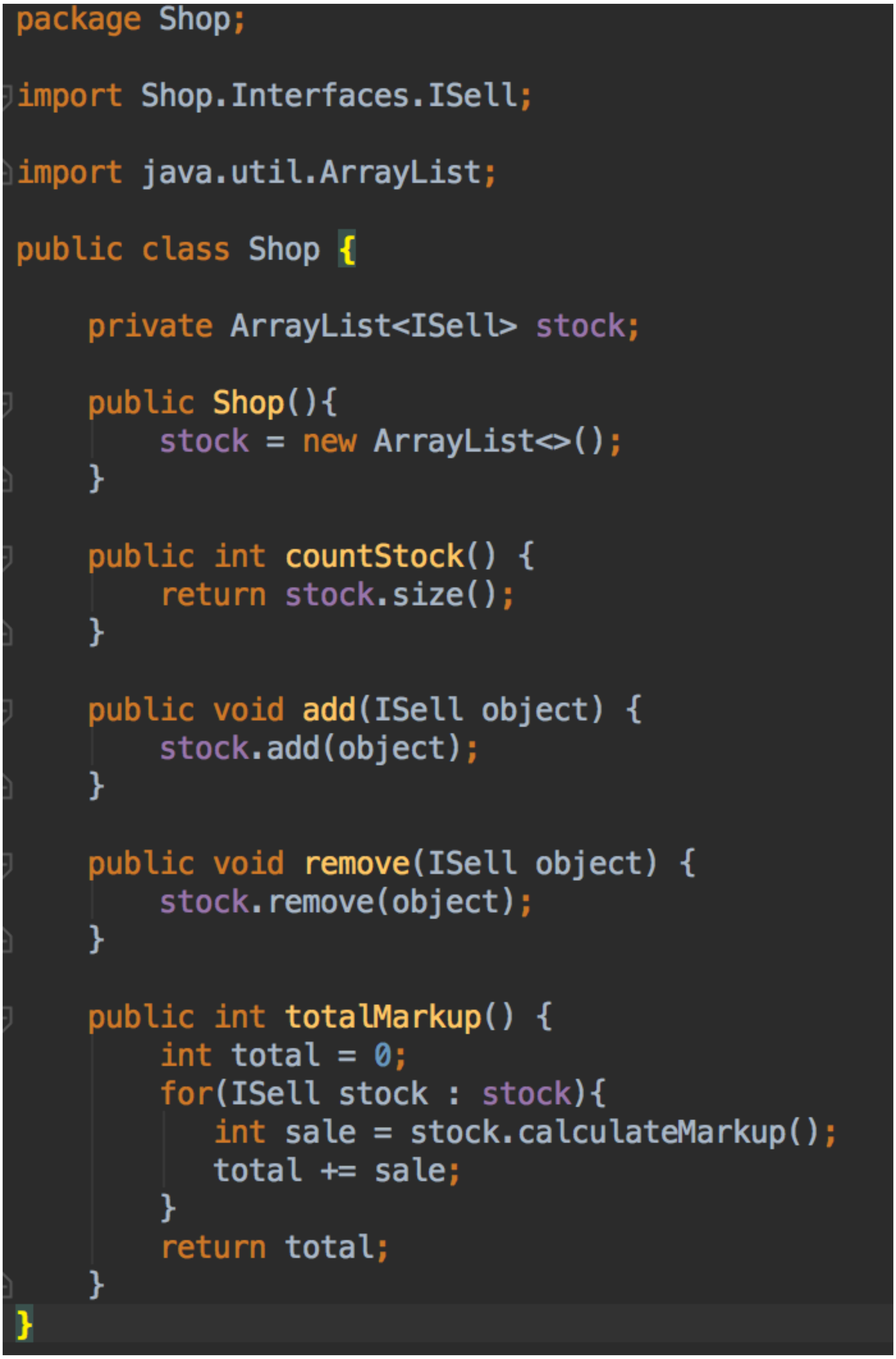
```java
package Shop.Interfaces;

public interface ISell {
    int calculateMarkup();
}
```



```java
package Shop.Instruments;

import Shop.Interfaces.IPlay;
import Shop.Interfaces.ISell;
import Shop.Stock;


public abstract class Instrument extends Stock implements IPlay, ISell{

    private InstrumentType type;

    public Instrument(InstrumentType type, String model, String make, int buyPrice, int sellPrice) {
        super(model, make, buyPrice, sellPrice);
        this.type = type;
    }


    public InstrumentType getType() {
        return type;
    }
}
```



```java
package Shop.Instruments;

public class Guitar extends Instrument {

    private int stringNumber;

    Guitar(InstrumentType type, String model, String make, int buyPrice, int sellPrice, int stringNumber) {
        super(type, model, make, buyPrice, sellPrice);
        this.stringNumber = stringNumber;
    }


    public int getStrings() {
        return stringNumber;
    }

    @Override
    public int calculateMarkup() {
        return getSellPrice()-getBuyPrice();
    }

    @Override
    public String play(){
        return getType() + " plays chord sound";
    }
}
```



```java
package Shop;

import Shop.Interfaces.ISell;

import java.util.ArrayList;

public class Shop {

    private ArrayList<ISell> stock;

    public Shop(){
        stock = new ArrayList<>();
    }

    public int countStock() {
        return stock.size();
    }

    public void add(ISell object) {
        stock.add(object);
    }

    public void remove(ISell object) {
        stock.remove(object);
    }

    public int totalMarkup() {
        int total = 0;
        for(ISell stock : stock){
            int sale = stock.calculateMarkup();
            total += sale;
        }
        return total;
    }
}
```