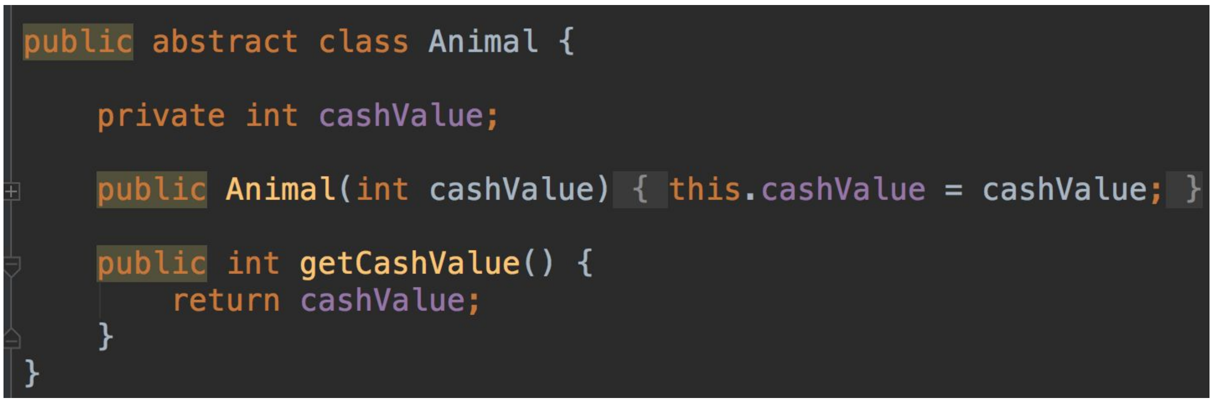
Evidence for Implementation and Testing Unit.

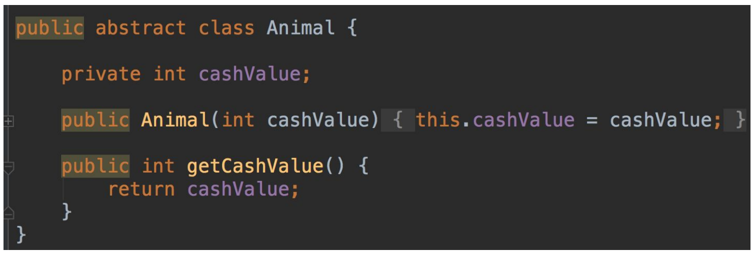
Yonatan Satat
E16
25/11/2017

I.T 1- Demonstrate one example of encapsulation that you have written in a program.

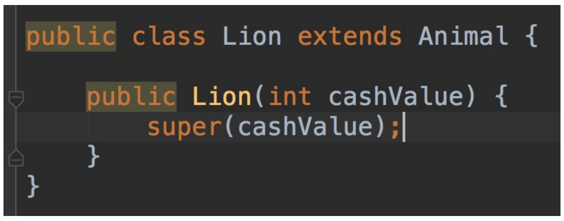


```java
public abstract class Animal {

    private int cashValue;

    public Animal(int cashValue) { this.cashValue = cashValue; }

    public int getCashValue() {
        return cashValue;
    }
}
```

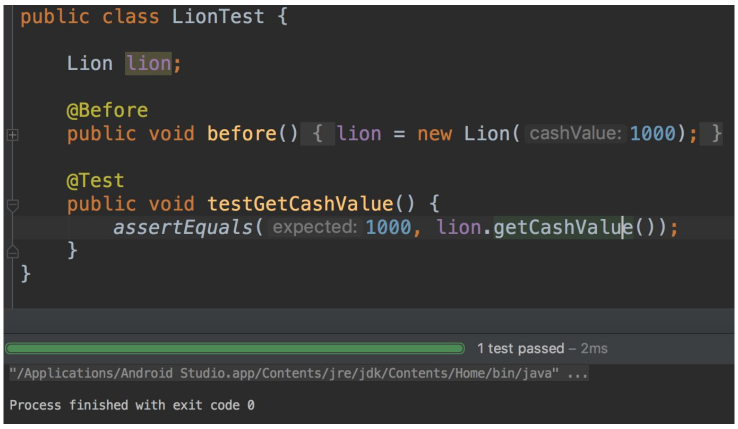
I.T 2 - Example the use of inheritance in a program.



```java
public abstract class Animal {

    private int cashValue;

    public Animal(int cashValue) { this.cashValue = cashValue; }

    public int getCashValue() {
        return cashValue;
    }
}
```



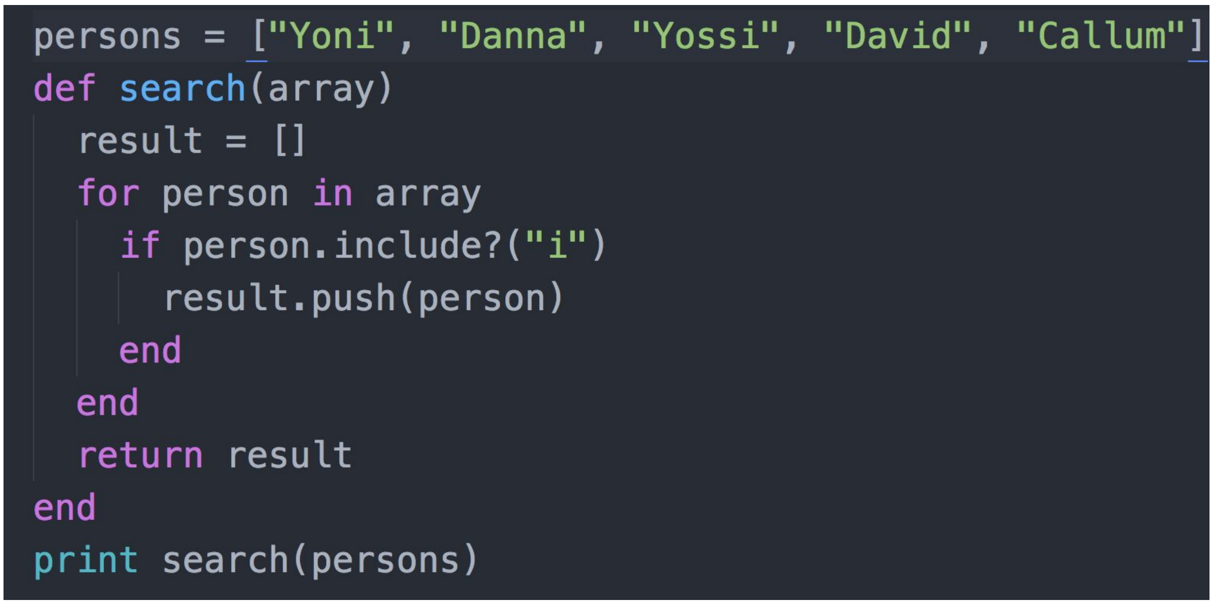
```java
public class Lion extends Animal {

    public Lion(int cashValue) {
        super(cashValue);
    }
}
```



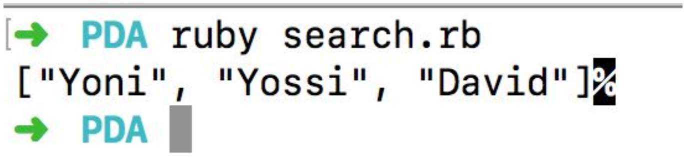
```java
public class LionTest {

    Lion lion;

    @Before
    public void before() { lion = new Lion( cashValue: 1000); }

    @Test
    public void testGetCashValue() {
        assertEquals( expected: 1000, lion.getCashValue());
    }
}
```

```
1 test passed – 2ms
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

Process finished with exit code 0
```

I.T 3 - Example of searching

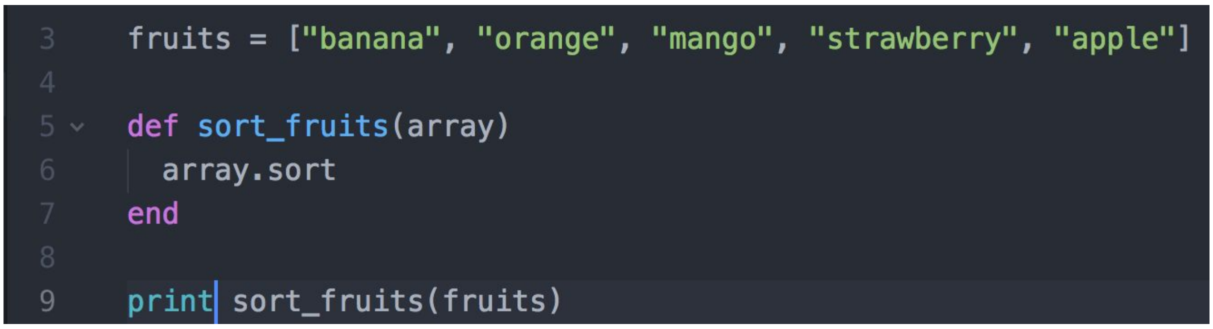


```
persons = ["Yoni", "Danna", "Yossi", "David", "Callum"]
def search(array)
  result = []
  for person in array
    if person.include?("i")
      result.push(person)
    end
  end
  return result
end
print search(persons)
```

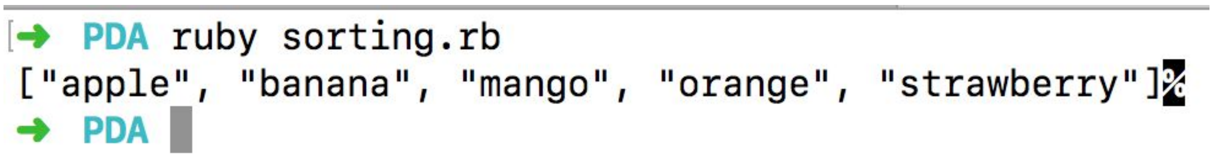


```
[➜  PDA ruby search.rb
["Yoni", "Yossi", "David"]%
➜  PDA
```

I.T 4 - Example of sorting

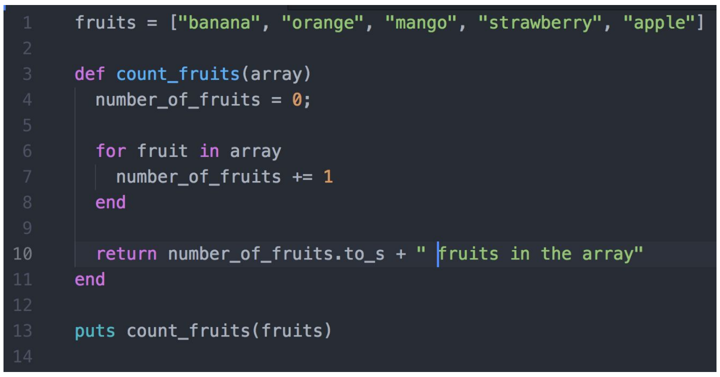


```
fruits = ["banana", "orange", "mango", "strawberry", "apple"]

def sort_fruits(array)
  array.sort
end

print sort_fruits(fruits)
```

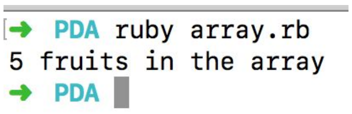


```
[➜  PDA ruby sorting.rb
["apple", "banana", "mango", "orange", "strawberry"]%
➜  PDA
```

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

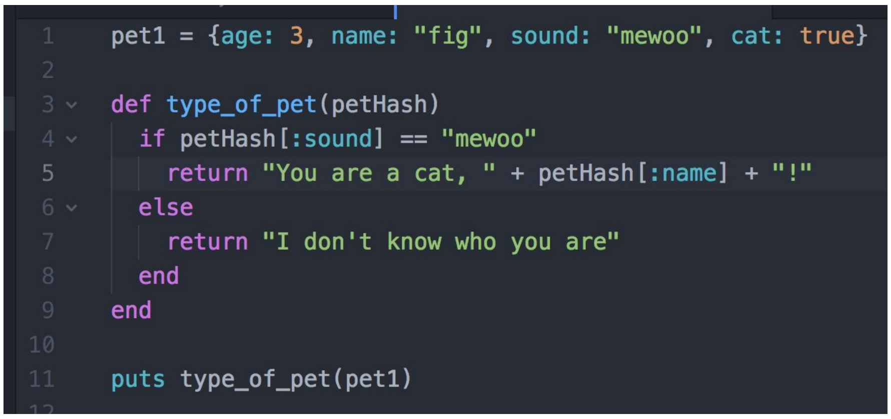


```ruby
fruits = ["banana", "orange", "mango", "strawberry", "apple"]

def count_fruits(array)
  number_of_fruits = 0;

  for fruit in array
    number_of_fruits += 1
  end

  return number_of_fruits.to_s + " fruits in the array"
end

puts count_fruits(fruits)
```

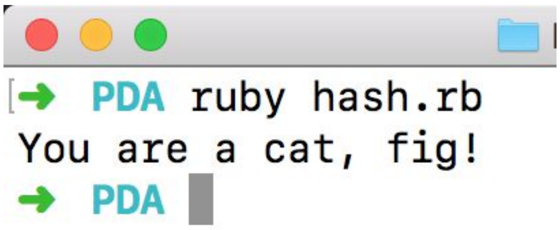


```
[➜  PDA ruby array.rb
5 fruits in the array
➜  PDA
```

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

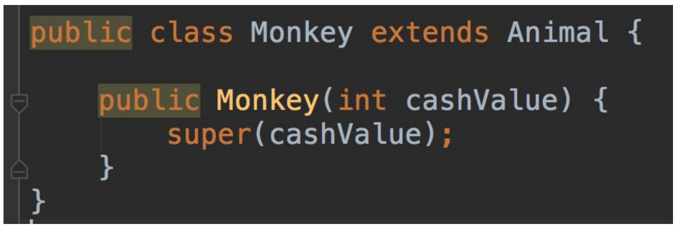


```ruby
pet1 = {age: 3, name: "fig", sound: "mewoo", cat: true}

def type_of_pet(petHash)
  if petHash[:sound] == "mewoo"
    return "You are a cat, " + petHash[:name] + "!"
  else
    return "I don't know who you are"
  end
end

puts type_of_pet(pet1)
```

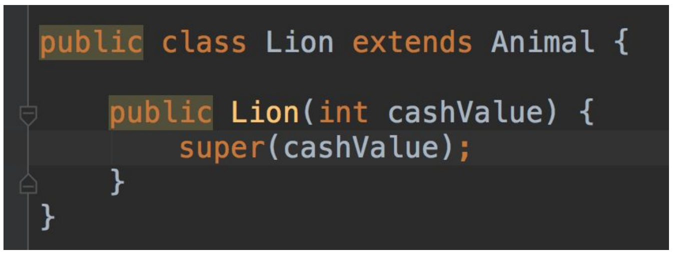


```
[➜  PDA ruby hash.rb
You are a cat, fig!
➜  PDA
```

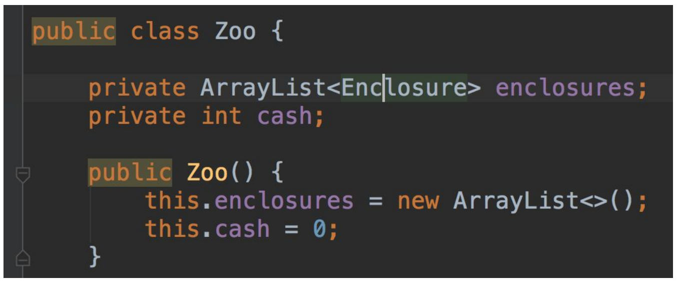
I.T 7 - Example of polymorphism in a program.



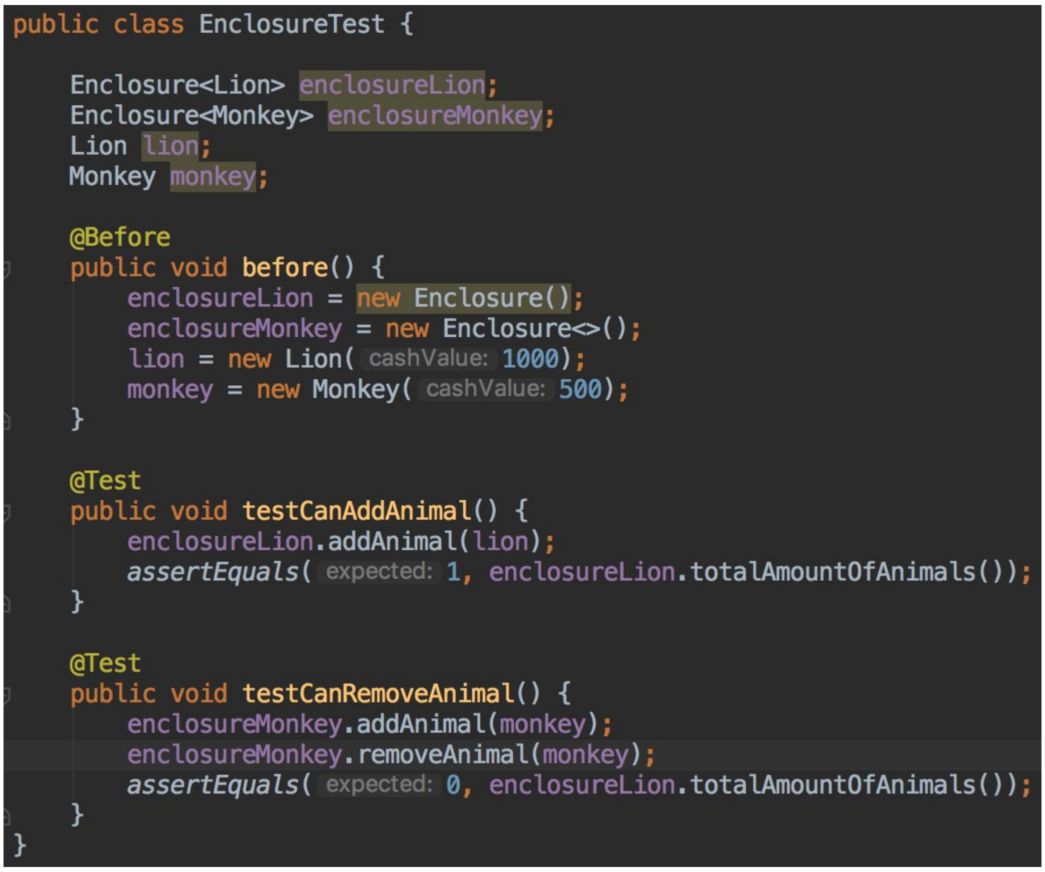
```java
public class Monkey extends Animal {

    public Monkey(int cashValue) {
        super(cashValue);
    }
}
```



```java
public class Lion extends Animal {

    public Lion(int cashValue) {
        super(cashValue);
    }
}
```



```java
public class Zoo {

    private ArrayList<Enclosure> enclosures;
    private int cash;

    public Zoo() {
        this.enclosures = new ArrayList<>();
        this.cash = 0;
    }
```



```java
public class EnclosureTest {

    Enclosure<Lion> enclosureLion;
    Enclosure<Monkey> enclosureMonkey;
    Lion lion;
    Monkey monkey;

    @Before
    public void before() {
        enclosureLion = new Enclosure();
        enclosureMonkey = new Enclosure<>();
        lion = new Lion( cashValue: 1000);
        monkey = new Monkey( cashValue: 500);
    }

    @Test
    public void testCanAddAnimal() {
        enclosureLion.addAnimal(lion);
        assertEquals( expected: 1, enclosureLion.totalAmountOfAnimals());
    }

    @Test
    public void testCanRemoveAnimal() {
        enclosureMonkey.addAnimal(monkey);
        enclosureMonkey.removeAnimal(monkey);
        assertEquals( expected: 0, enclosureLion.totalAmountOfAnimals());
    }
}
```