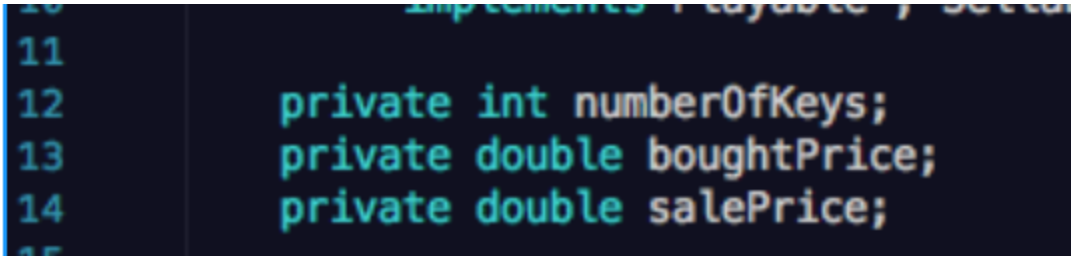
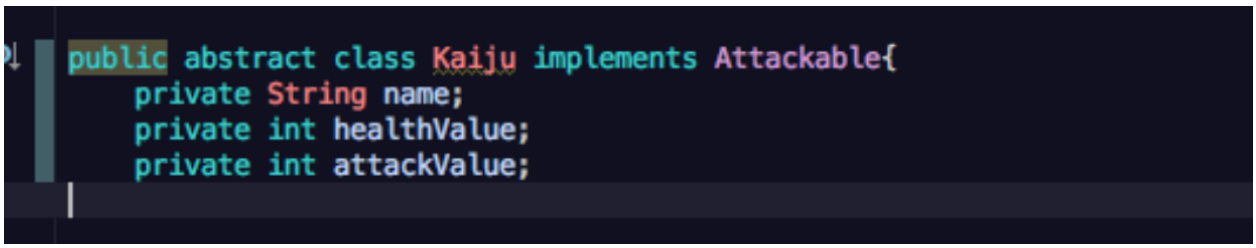
Allyson Wilson
Cohort 15
16/8/17

I.T 1 Encapsulation

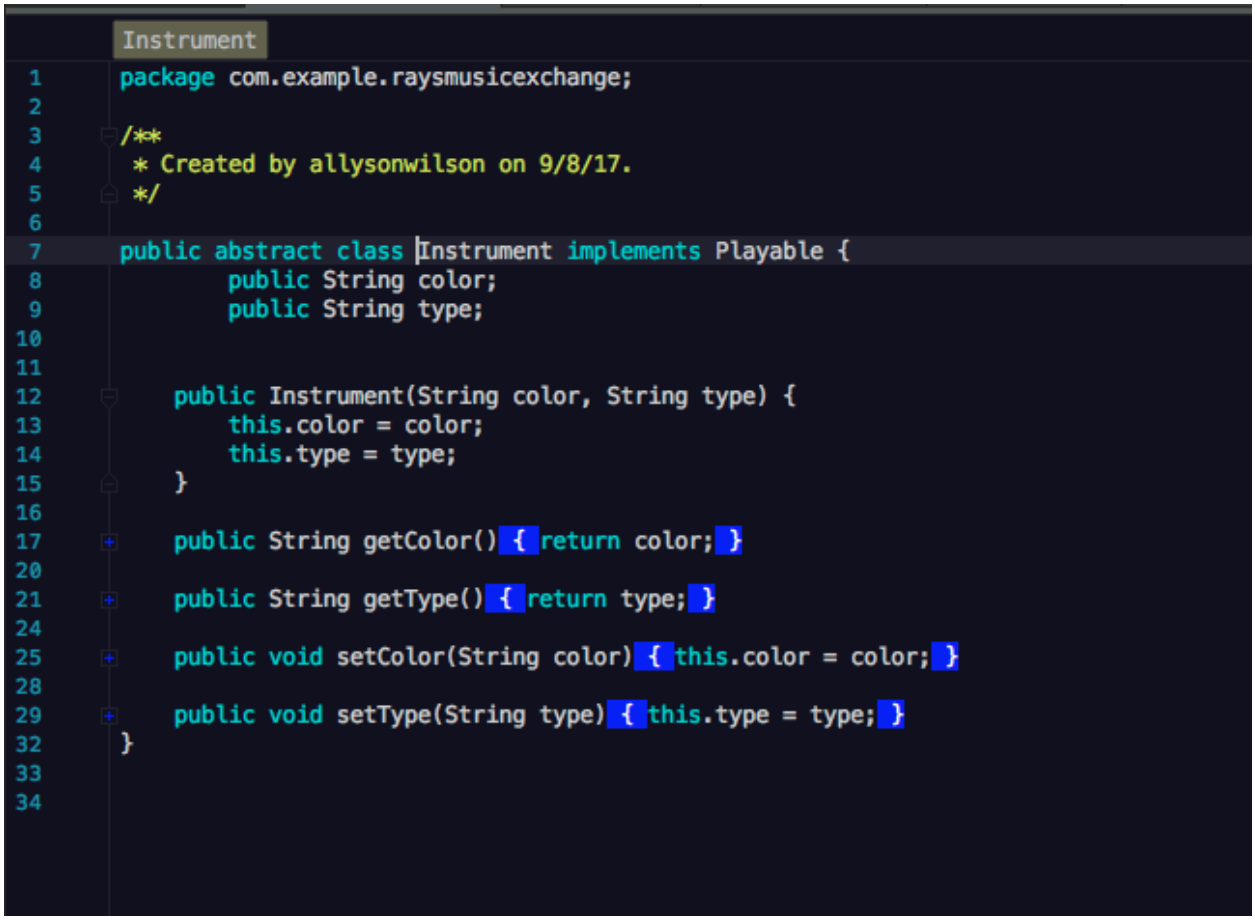


```
11
12          private int numberOfKeys;
13          private double boughtPrice;
14          private double salePrice;
```

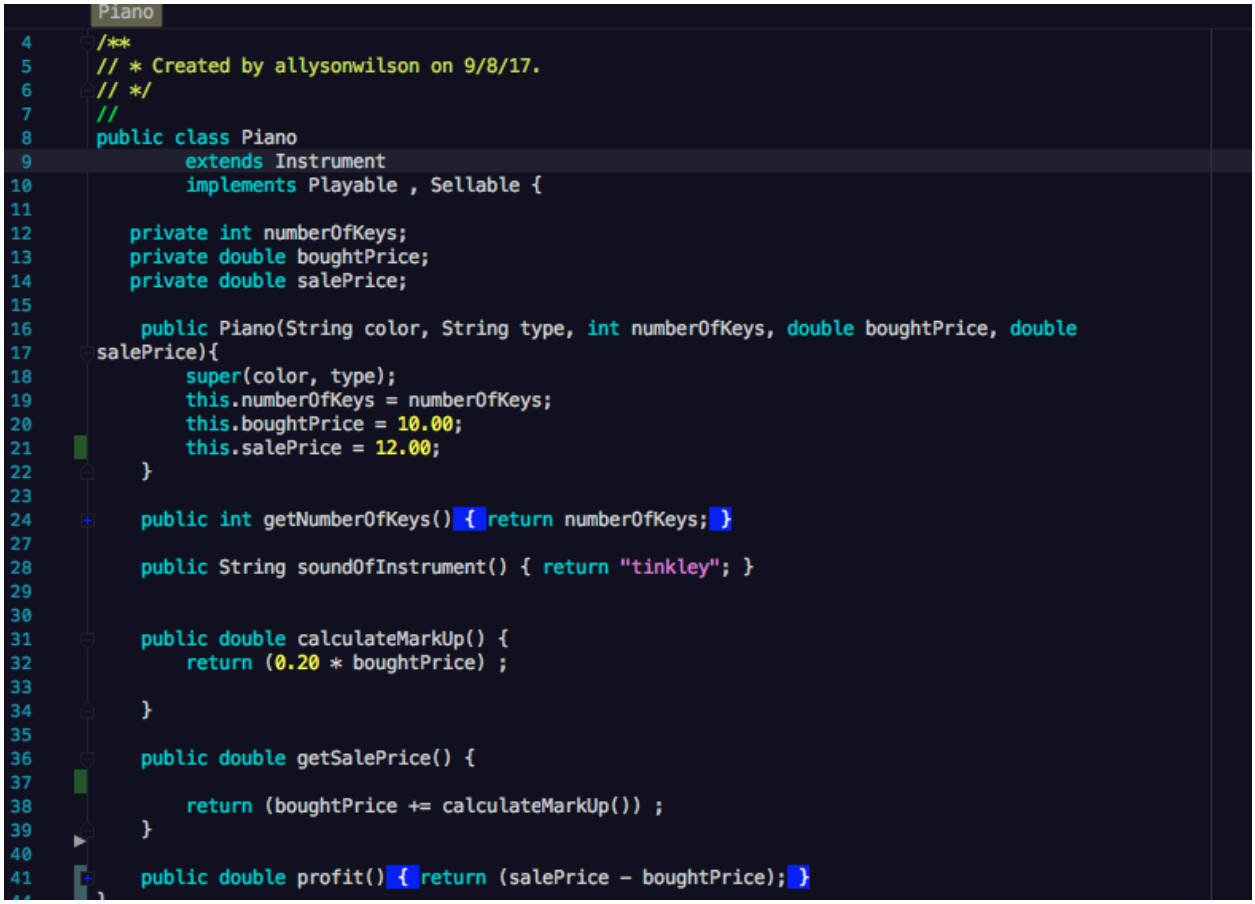


```
public abstract class Kaiju implements Attackable{
    private String name;
    private int healthValue;
    private int attackValue;
```

I.T 2 Superclass



I.T 2 Class inherits from previous class



I.T 2 Object in Inherited Class

```java
import ...
/**
 * Created by allysonwilson on 9/9/17.
 */

public class PianoTest {

    Piano piano;

@Before
    public void before() { piano = new Piano( "white", "string instrument", 88, 10.00, 12.00); }

    @Test
    public void testHasColor() { assertEquals( "white", piano.getColor() ); }

    @Test
    public void testHasType() { assertEquals( "string instrument" , piano.getType() ); }

    @Test
    public void testHasNumberOfKeys() { assertEquals( 88, piano.getNumberOfKeys() ); }

    @Test
    public void testSoundOfInstrument() { assertEquals( "tinkley" , piano.soundOfInstrument()); }

    @Test
    public void testCalculateMarkUp() { assertEquals( 2.00, .01, piano.calculateMarkUp() ); }

    @Test
    public void testHasSalePrice() { assertEquals(12.00, 0.1, piano.getSalePrice() ); }

    @Test
    public void testHasProfit() { assertEquals( 2.00, 0.01, piano.profit() ); }
}
```

I.T 2 Method from inherited class of Sellable

```java
    public double calculateMarkUp() {
        return (0.20 * boughtPrice) ;

    }
```

IT 3 Search Data

```ruby
# adoption.rb
  end

  def self.all()
    sql = "SELECT * FROM adoptions"
    values = []
    results = SqlRunner.run( sql, values )
    return results.map { |adoption| Adoption.new( adoption ) }
  end

  def animal()
    sql = "SELECT * FROM animals
    WHERE id = $1"
    values = [@animal_id]
    results = SqlRunner.run( sql, values )
    return Animal.new( results.first )
  end

  def hero()
    sql = "SELECT * FROM heroes
    WHERE id = $1"
    values = [@hero_id]
    results = SqlRunner.run( sql, values )
    return Hero.new( results.first )
  end
```

Result of function running

```
Shelter_Project — ruby db/seeds.rb — ruby — ruby db/seeds.rb — 113×32
[2] pry(main)> adoption1.hero()
=> #<Hero:0x007ffd4c2b34b8 @id=4, @image="/images/Ona_avatar.png", @name="Ona">
[3] pry(main)> hero1.name()
=> "Ona"
[4] pry(main)> animal1.species()
=> "Basking Shark"
[5] pry(main)>
```

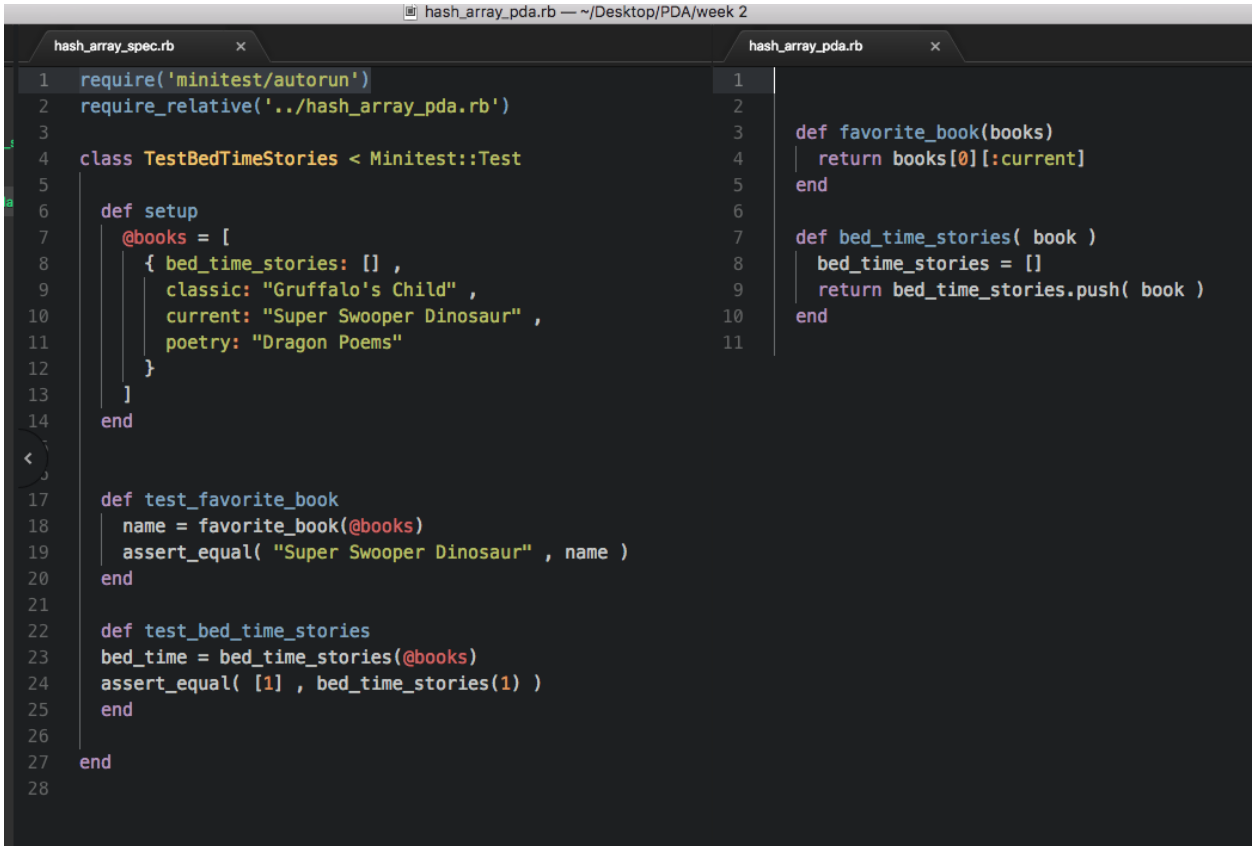
IT 4 Sort Data

```ruby
def self.all()
  sql = "SELECT * FROM heroes ORDER by name"
  values = []
  results = SqlRunner.run( sql, values )
  return results.map { |hash| Hero.new( hash ) }
end
```

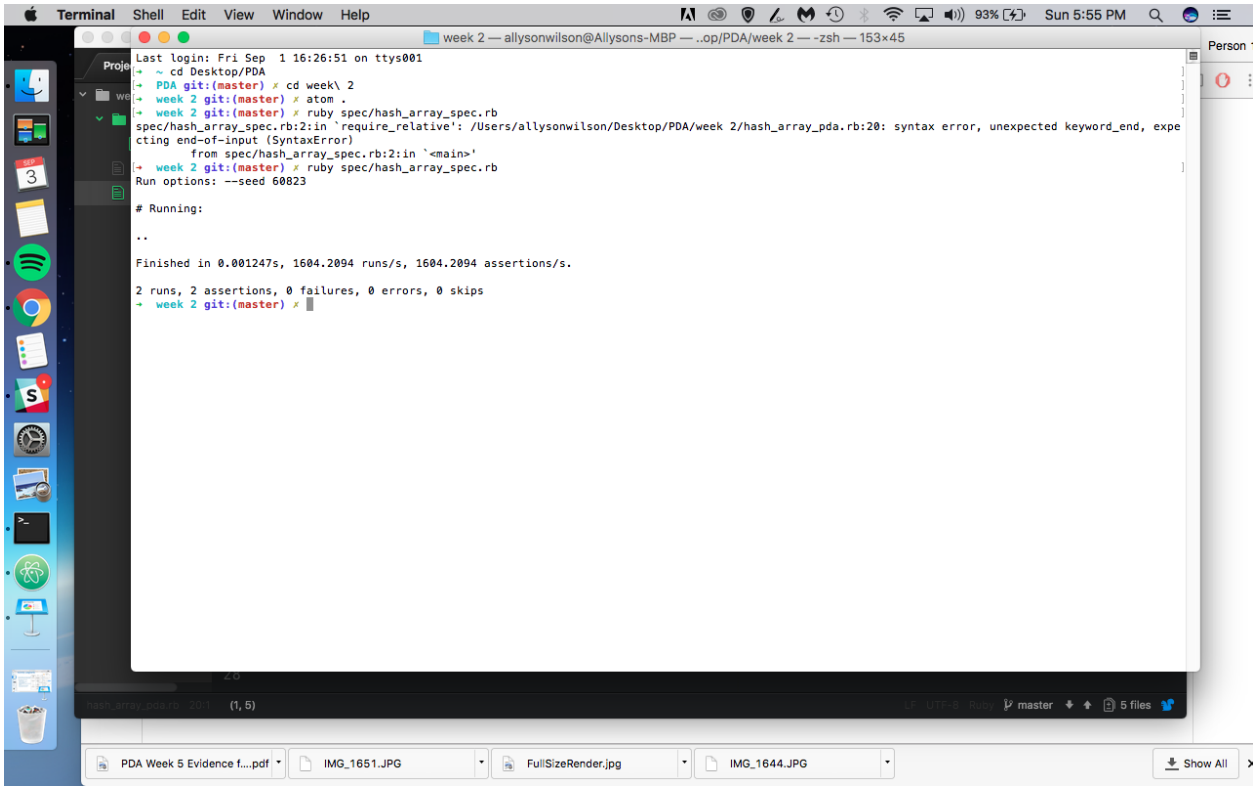
```ruby
def self.all()
  sql = "SELECT * FROM heroes ORDER by name"
  values = []
  results = SqlRunner.run( sql, values )
  return results.map { |hash| Hero.new( hash ) }
end
```

```
→  Shelter_Project git:(master) ✗ ruby db/seeds.rb

From: /Users/allysonwilson/Desktop/Shelter_Project/db/seeds.rb @ line 62 :

    57:   'hero_id' => hero1.id
    58: })
    59: adoption1.save()
    60:
    61: binding.pry
 => 62: nil

[1] pry(main)> Hero.all()
=> [#<Hero:0x007fc97183cc68 @id=8, @image="/images/Granny.png", @name="Granny">,
 #<Hero:0x007fc971836ca0 @id=9, @image="/images/Fraser_avatar.png", @name="Joe">,
 #<Hero:0x007fc971835aa8 @id=7, @image="/images/Ona_avatar.png", @name="Ona">]
[2] pry(main)>
```

IT 5 and IT 6 Hash and Array with TDD



Terminal Print out of successful tests below:



IT 7 Polymorphism in a program