Greenfoot Chapter 2 Exercises

2.1: Upon pressing the run button the crab stays in place and nothing moves at all.

2.2:

```
public void act()
   {
      move(5);
   }
```

2.3: Increasing the number increases the speed of the crabs, I assume it means the number of steps taken per increment and making it 50 instead of 5 makes them appear to move faster because they take 50 straps each time.

2.4: Placing multiple crabs and running the programs shows that all crabs are affected in the same way, and that they all move at the same speed.

2.5: Changing the move value to a negative number causes the crab to move to the left.

2.6 :

```
public void act()
   {
      turn(30);
   }
```

2.7: Changing the value to a negative one cause the crab to rotate counter-clockwise or left.

2.8:

```
public void act()
   {
      turn(90);
      move(50);
   }
```

2.9:

```
public void act()
{
    turn(90);
    move(50)
}
```

Runs into: Error: ; expected

2.10: Removing the i in void results in *cannot find symbol class vod*

Removing the parentheses after act results in *illegal start of type*

Removing a curly brace results in *reached and of file while parsing*

Removing the t in extends resulted in *{ expected*

Removing the a in class prompts many errors all of which state: *class, interface, or enum expected*

2.11: The actor class has 25 different methods

2.12: the isAtEdge() method seems like it would be useful in detecting when the crab is at the edge of the screen.

2.13: Running the isAtEdge() command from the middle of the screen results in a return false.

2.14: Running the command again while the crab is at the edge of the world results in a return true.

2.15:

```
public void act()
  {
    if( isAtEdge())
    {
      turn(17);
    }
    move(5);

}
```

2.16: I think turn(180) is a good command.

2.17: Putting the move statement into the if statement ‘breaks’ the code because since the crab can no longer move unless it is fulfilling the at edge condition, it cannot get to the edge to complete the condition.

2.18: Removing some of the curly braces changes to color based on where the bracket began, so the if statement not having a closing bracket makes the ensuing code enveloped in it and changes the color to purple. Or leaving a bracket from the opening declaration changed it to green.

2.19: 1. Public void play();
a. Play
b. No
c. 0
2. Public void addAmount(int amount);
a.addAmount
b.no
c.1
3.public boolean hasWings();
a.hasWings
b.yes, boolean
c.0
4. Public void compare(int x, int y, int z)
A.compare
B.no
C.3
5.public boolean isGreater(int number);
a.isGreater
B.yes, boolean
C.1

2.20: public void go();

2.21: public int process(int number);

2.22: public boolean isOpen();

```
2.23: play();
      compare(1, 3, 5);
      addAmount(20);
```

2.24: 12 methods

2.25: 5 methods

2.26: 1 parameter

2.27: I used the dance methos and the cat started dancing to stayin’ alive by the beegees.

2.28: The cat is bored, but is not bored while dancing.

2.29: I opened the cat editor

2.30:

```
public void act()
  {
     eat();
  }
```

2.31:

```
public void act()
  {
     dance();
  }
```

2.32:

```
public void act()
  {
     sleep(3);
  }
```

Cat will sleep for 3 cycles

```
2.33: public void act()
    {
        dance();
        eat();
        sleep(10);
    }

2.34: public void act()
    {
        if( isSleepy())
        {
            sleep(3);
        }
    }

2.35: public void act()
    {
        if( isSleepy())
        {
            sleep(3);
        }
        if (isBored())
        {
            dance();
        }
    }


2.36: public void act()
    {
        if( isSleepy())
        {
            sleep(3);
        }
        if (isBored())
        {
            dance();
        }
        if( isHungry())
        {
            eat();
        }
    }
```

```
2.37: public void act()
    {
        if( isSleepy())
        {
            sleep(3);
        }
        else
        {
            shoutHooray();
        }
        if (isBored())
        {
            dance();
        }
        if( isHungry())
        {
            eat();
        }
    }
```

Can make the cat tired by invoking the dance method.

```
2.38: public void act()
    {
```

```
    if( isSleepy())
    {
        sleep(3);
    }
    else
    {
        shoutHooray();
    }
    if (isBored())
    {
        dance();
    }
    if( isHungry())
    {
        eat();
    }
    if( isAlone())
    {
        sleep(2);
    }
    else
    {
        shoutHooray();
    }
}
```