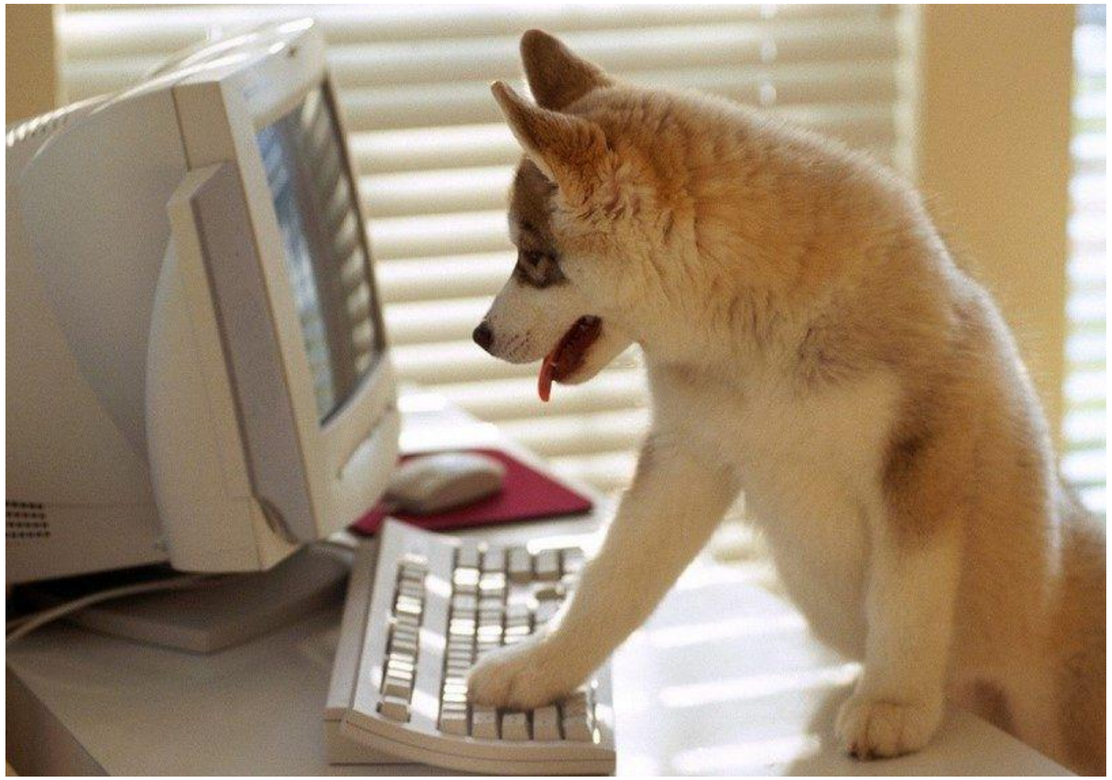
Java Programming



Organizational Stuff

18.03.: Structures
19.03.: Methods
20.03.: Recursion
21.03.: Arrays
22.03.: Strings

25.03.: OOP1
26.03.: OOP2
27.03.: Generics
28.03.: Exceptions & Enums
29.03.: GUI

Recursion

Something defined by itself

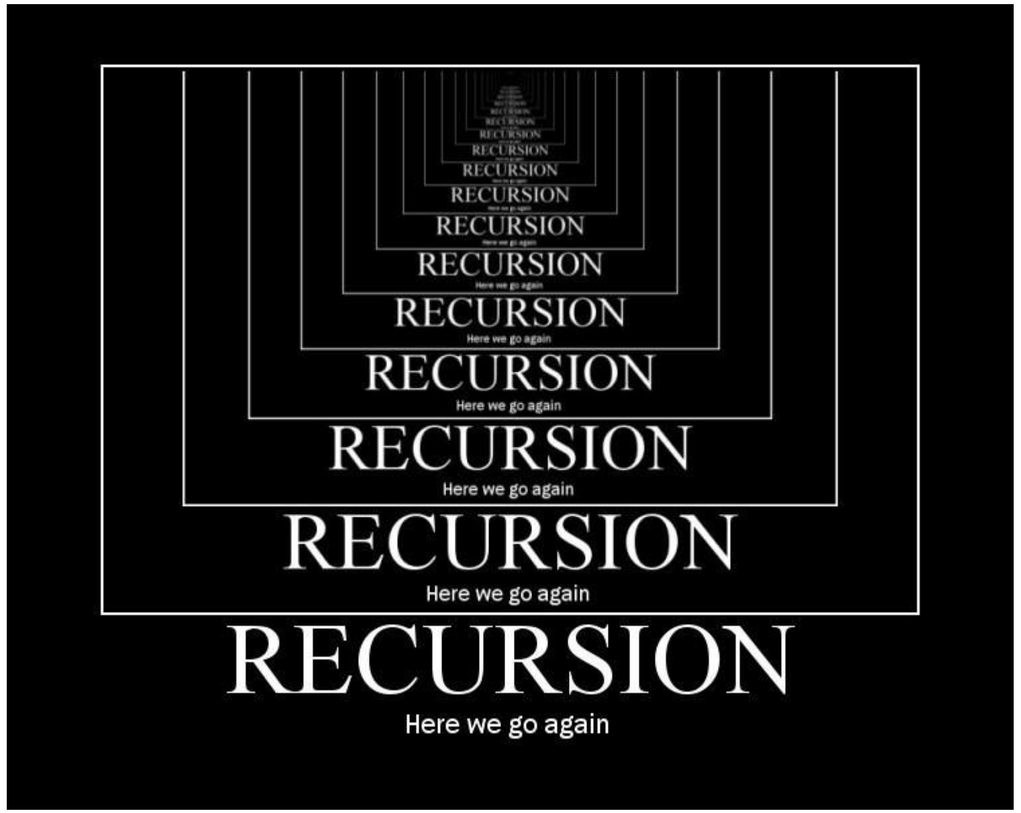


Image: https://cdn-images-1.medium.com/max/1600/1*appBwh6_RtvocVxwqpplHA.jpeg

Recursion

```
public static int rekExample(int i){
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);
  }
}
```

Recursion

```
public static int rekExample(int i){   //x=3
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);
  }
}
```

Recursion

```
public static int rekExample(int i){   //x=3
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);
  }
}
```

Recursion is fun!

Recursion

```
public static int rekExample(int i){   //x=3
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);    //x=2
  }
}
```

Recursion is fun!

Recursion

```
public static int rekExample(int i){   //x=2
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);    //x=2
  }
}
```

Recursion is fun!

Recursion

```
public static int rekExample(int i){   //x=2
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);    //x=2
  }
}
```

Recursion is fun!
Recursion is fun!

Recursion

```
public static int rekExample(int i){   //x=2
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);    //x=1
  }
}
```

Recursion is fun!
Recursion is fun!

Recursion

```
public static int rekExample(int i){   //x=1
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);    //x=1
  }
}
```

Recursion is fun!
Recursion is fun!

Recursion

```
public static int rekExample(int i){   //x=1
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);    //x=1
  }
}
```

Recursion is fun!
Recursion is fun!
Recursion is fun!

Recursion

```
public static int rekExample(int i){   //x=1
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);    //x=0
  }
}
```

Recursion is fun!
Recursion is fun!
Recursion is fun!

Recursion

```
public static int rekExample(int i){   //x=0
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);    //x=0
  }
}
```

Recursion is fun!
Recursion is fun!
Recursion is fun!

Recursion

```
public static int rekExample(int i){   //x=0
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);    //x=0
  }
}
```

Recursion is fun!
Recursion is fun!
Recursion is fun!

Recursion

```
public static int rekExample(int i){
  if(x<1){
    return x;
  }
  else{
    System.out.println("Recursion is fun!");
    return rekExample(x-1);
  }
}
```

Recursion

Today's Assignment:
https://classroom.github.com/a/U6vstOSW