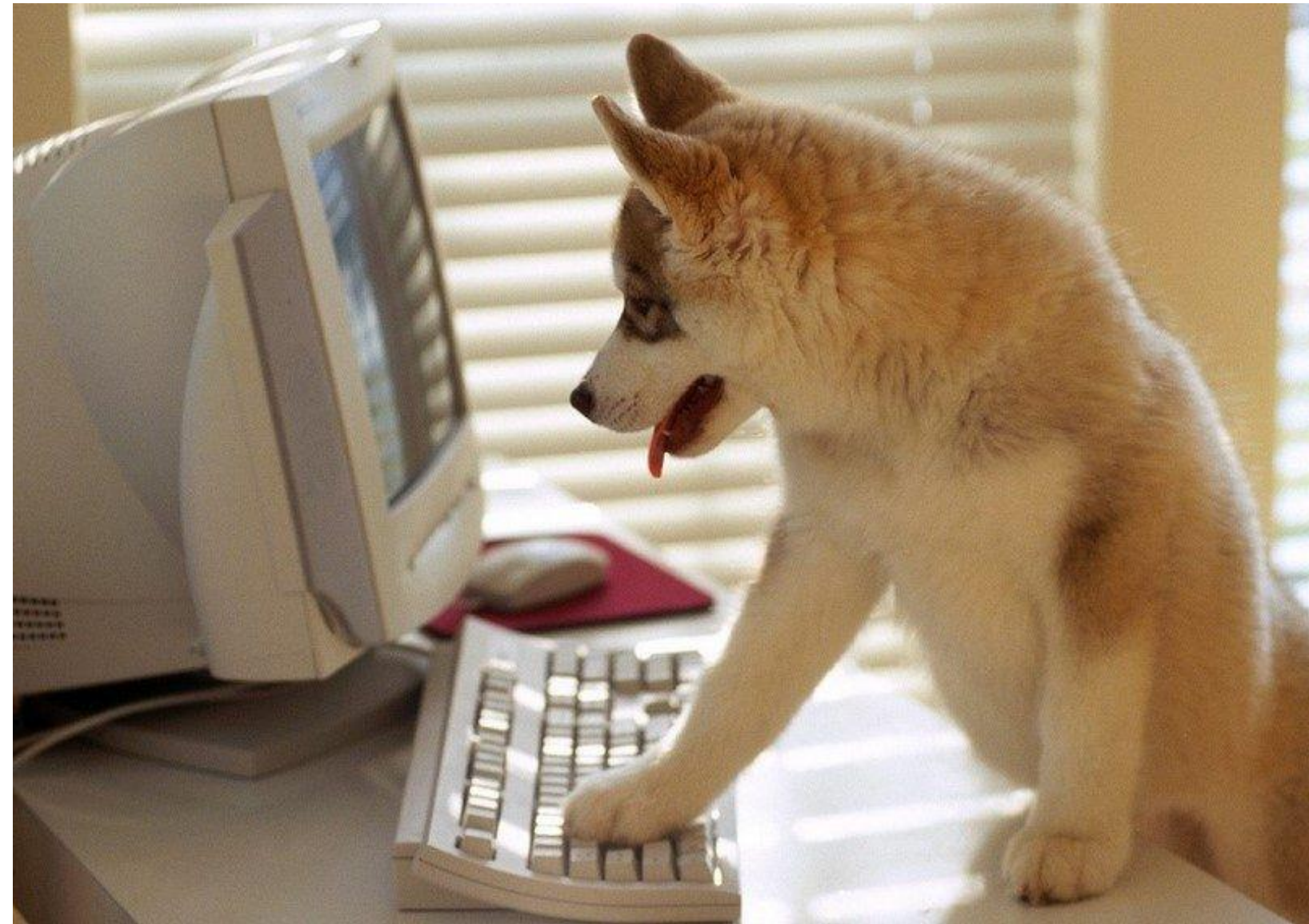


Java Programming



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

Style Conventions



Image Source:

<https://memegenerator.net/instance/76894967/god-kills-a-kitten-every-time-you-violate-code-style-god-kills-a-kitten>

Intendation

Good:

```
public class GoodCode{
    public static void main(String[]
                               args){
        Scanner scan = new
                               Scanner(System.in);
        int a = scan.nextInt();
        int b = scan.nextInt();

        if(a>b){
            System.out.println(a);
        }
        else{
            System.out.println(b);
        }
    }
}
```

Bad:

```
public class BadCode{
    public static void main(String[] args){
        Scanner scan = new Scanner(System.in);
        int a = scan.nextInt();
        int b = scan.nextInt();
        if(a>b){
            System.out.println(a);
        }
        else{
            System.out.println(b);
        }
    }
}
```

Naming

Good:

```
public class GoodCode{
    public static void main(String[]
                           args) {

        Scanner scan = new
                        Scanner(System.in);
        double width = scan.nextDouble();
        double height = scan.nextDouble();

        double area = computeArea(width,
                                   height);

        System.out.println(area);

    }
}
```

Bad:

```
public class BadCode{
    public static void main(String[]
                           args) {

        Scanner scan = new
                        Scanner(System.in);
        double a = scan.nextDouble();
        double a2 = scan.nextDouble();

        double a3 = check3(a, a2);

        System.out.println(a3);

    }
}
```

Arrays

Storing multiple values in one Variable

But: Length and Datatype are fixed!

Arrays

```
int [] arr = new int[3];  
arr[0] = 1;  
arr[1] = 2;  
arr[2] = 3;
```

Arrays

```
int [] arr = new int[3];  
arr[0] = 1;  
arr[1] = 2;  
arr[2] = 3;
```

```
int [] arr2 = {1,2,3};
```


Arrays

```
int [] arr = {1,2,3};

//iterates over the indices
for(int i = 0; i<arr.length; i++){
    System.out.println(arr[i]);
}

//iterates over the values
for(int i : arr){
    System.out.println(i);
}
```

2D Arrays

```
int [][] arr = new int[3][2];  
arr[0][0] = 1;  
arr[0][1] = 2;  
arr[1][0] = 3;  
arr[1][1] = 4;  
arr[2][0] = 5;  
arr[2][1] = 6;
```

```
int [][] arr2 = {{1,2},  
                 {3,4},  
                 {5,6}};
```

```
int [][] arr3 = new int[3][];
```

2D Arrays

```
int [][] arr = {{1,2},
                {3,4},
                {5,6}};

//iterates over the indices
for(int i = 0; i<arr.length; i++){
    for(int j = 0; j<arr[i].length; j++){
        System.out.println(arr[i][j]);
    }
}

//iterates over the values
for(int[] i : arr){
    for(int j : i){
        System.out.println(j);
    }
}
```

Recursion

Today's Assignment:

<https://classroom.github.com/a/bz5ZEiVh>

