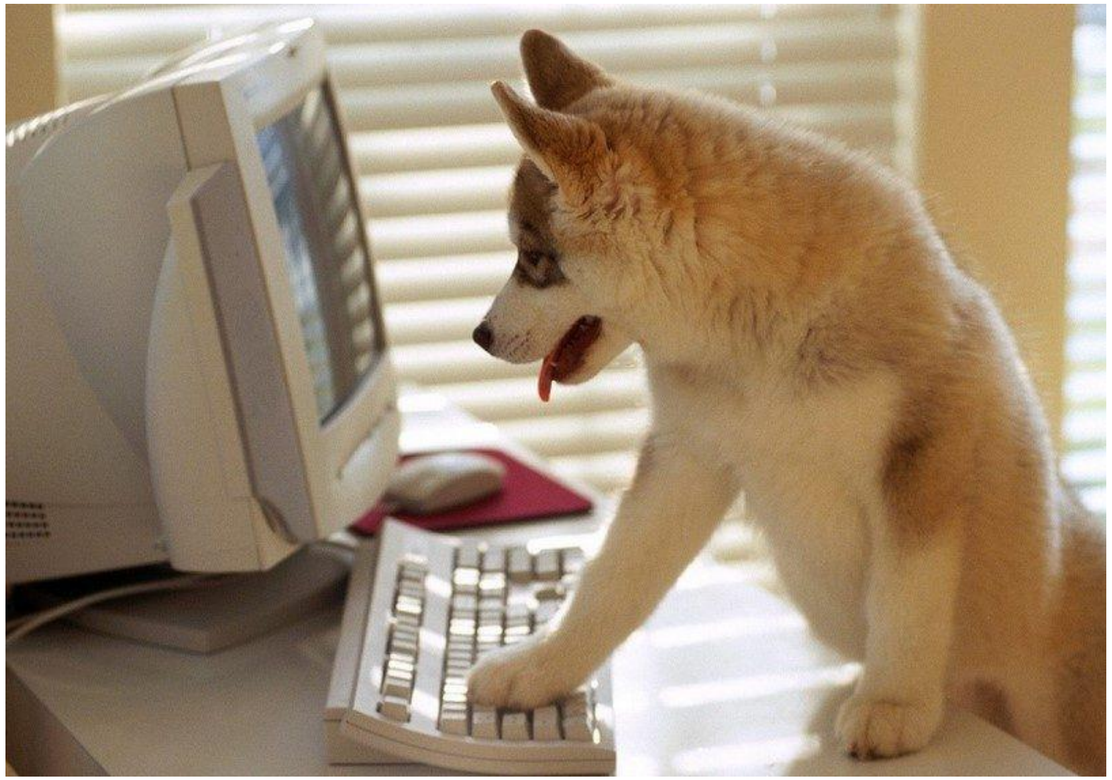
Java Programming



Organizational Stuff

23.09.: No Students :(
24.09.: Structures
25.09.: Methods & Recursion
26.09.: Arrays
27.09.: Strings

30.09.: OOP
01.10.: Generics & Linked Lists
02.10.: Exceptions & Testing
03.10.: Holiday
04.10.: GUI

Generics

Java is very strict about data types.
But what if we don’t know the data type or want to keep it flexible?

Generics

```
public class Example<T>{
   private T data;
   public Example(T data){
      this.data=data
   }
   public T getData(){
      return data;
   }
}
```

Generics

```
public class Example<T>{
   private T data;
   public Example(T data){
      this.data=data
   }
   public T getData(){
      return data;
   }
}
```

Generics

```
public class Example<T>{
   private T data;
   public Example(T data){
      this.data=data
   }
   public T getData(){
      return data;
   }
}
```

```
Example<String> ex = new Example<>(“Hello”);
System.out.println(ex.getData());
```

Generics

```
public class Example<T>{
   private T data;
   public Example(T data){
      this.data=data
   }
   public T getData(){
      return data;
   }
}
```

```
Example<String> ex = new Example<>(“Hello”);
System.out.println(ex.getData());
```

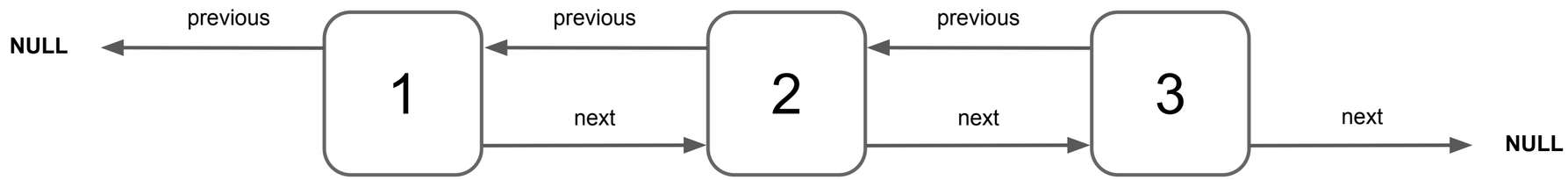
Generics

Real World Scenarios?

- ArrayLists
- HashMaps
- Collections
- ...

Generics

Example: Double Linked Lists



Generics

Today's Assignment:
https://classroom.github.com/a/K6ER4dEj