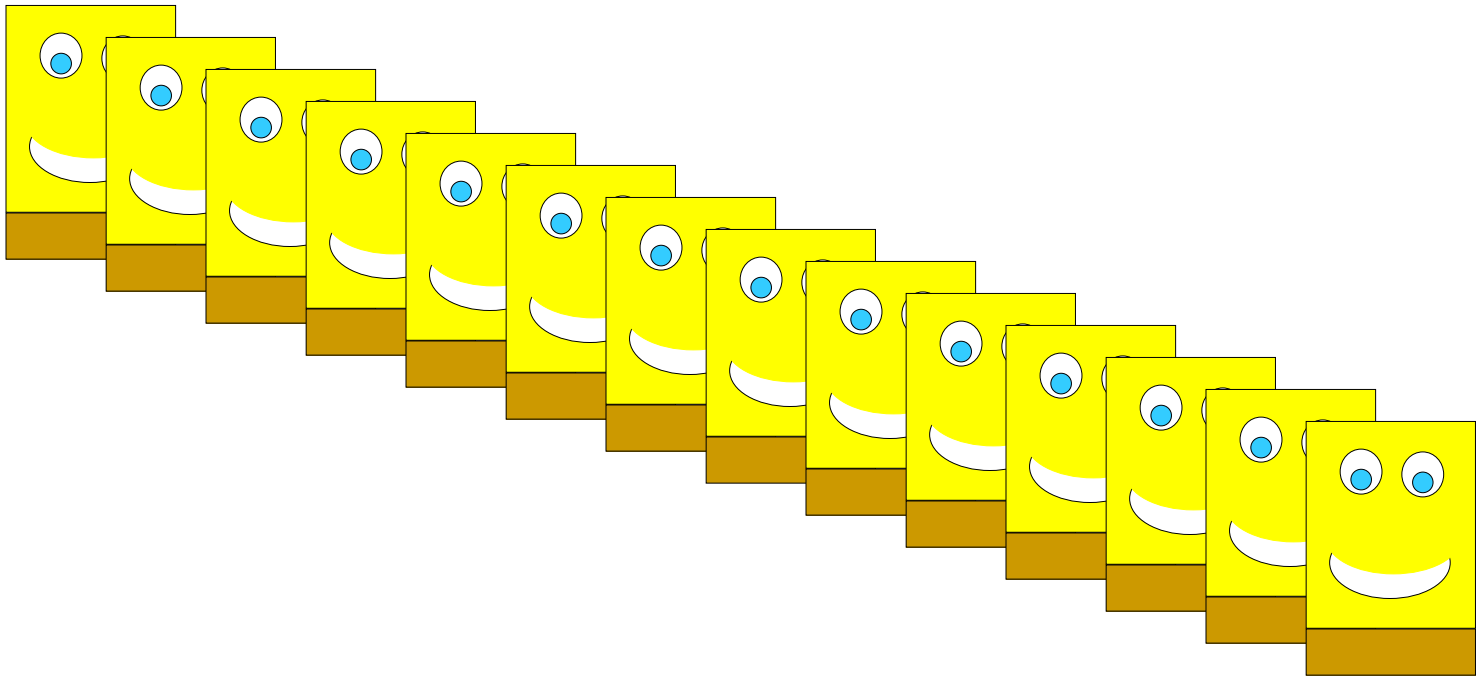
Something new: ArrayList Class

Mr. Neat

Java

What are the limitations of an array?



Limitations of arrays

Cannot be resized!

java.util.ArrayList is the library class that must be imported.

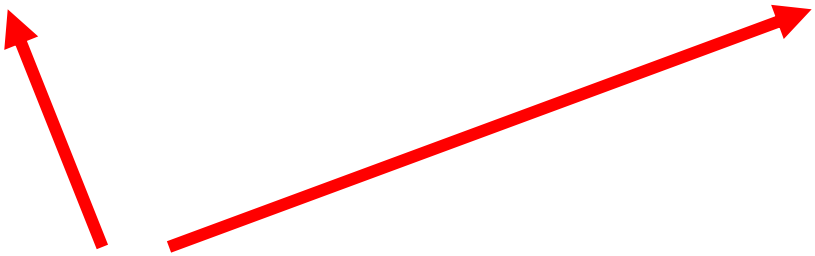
import java.util.ArrayList;

```
public class PlayArray
{
        private ArrayList<Object> myList = new ArrayList<Object>();
```

constructs an ArrayList object
with zero elements

import java.util.ArrayList;

```
public class PlayArray
{
        private ArrayList<Object> myList = new ArrayList<Object>();
```



can put any class in here

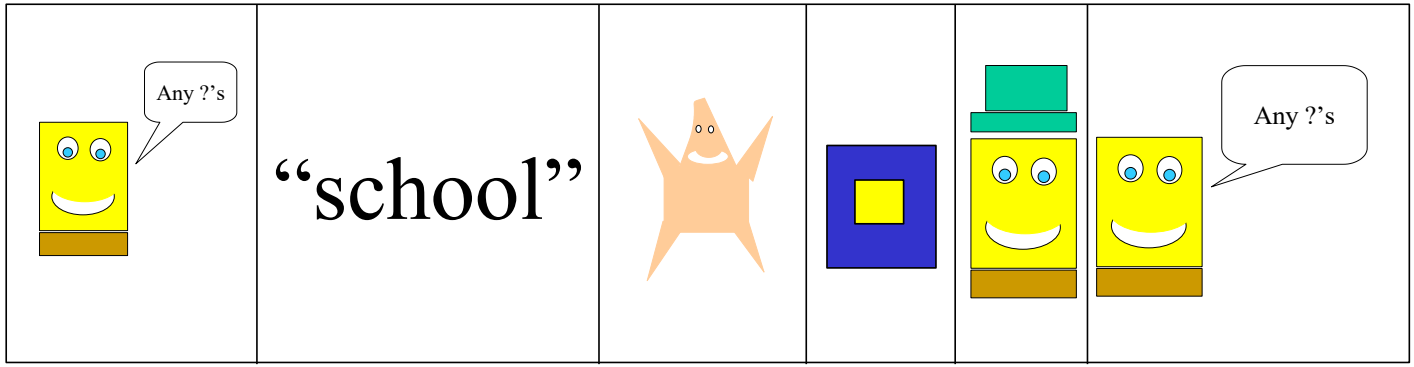
constructs an ArrayList object
with zero elements

public ArrayList Methods

int size()	myList.size()
boolean add(Object x)	myList.add(“Hi”);
Object get(int index)	myList.get(4);
Object set(int index, Object x)	myList.set(3,”Bye”);
Object remove(int index)	myList.remove(2);
void add(int index, Object x)	myList.add(3,new SpongeBob…

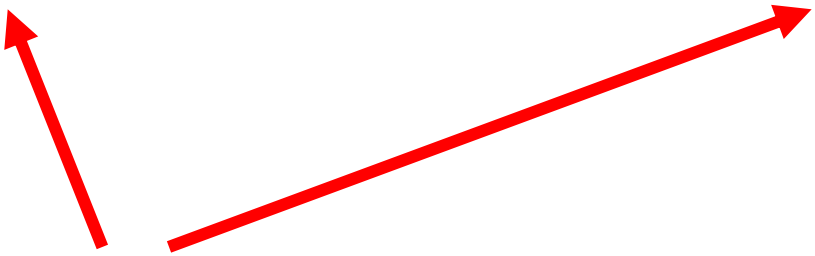
---

myList



import java.util.ArrayList;

```
public class PlayArray
{
        private ArrayList<String> myList2 = new ArrayList<String>();
```



can put any class in here

constructs an ArrayList object
with zero elements

public ArrayList Methods

int size()	myList2.size()
boolean add(String x)	myList2.add(“Hi”);
String get(int index)	myList2.get(4);
String set(int index, String x)	myList2.set(3,”Bye”);
String remove(int index)	myList2.remove(2);
void add(int index, String x)	myList2.add(3,new SpongeBob….);

---

myList2 (holds Strings)

“hello”	“school”	“Hi”	“Bye”	“I”

Don’t Forget the for each loop!

```
for(String s: myList2)
{
    SOP(s);
}
```

Don’t Forget the for each loop!

```
for(String s: myList2)
{
    SOP(s);
}
```

Simpler notation – no need to call .get(i)

How does the ArrayList resize?

One problem…
what if we wanted an ArrayList of ints, booleans or doubles (primitives)?

But ArrayList holds Classes!

Wrapper Classes to the rescue!

These classes turn primitive types into classes. We are concerned about 3:

Double

Int

Boolean

Construct a Double object:

```
Double r = new Double(7.5);

// to retrieve the double value,…

System.out.print(r.doubleValue());
```

Let's add a Double value to aList:

```
ArrayList<Double> aList = new ArrayList<Double>();
double num = 37.5;
Double numWrap = new Double(num);
aList.add(numWrap);
```

How would you retrieve the Double in aList?

Now its an Object….

```
Double retriever = aList.get(0);
double back = retriever.doubleValue();
```

Lab

- Make an ArrayList of Cars
- Store 100 Cars in the ArrayList
- Make them recycle