ShapesHandling

```
//1
public class Circle {
String name;
double x,y,r;

public Circle (String st) {
name = st;
}
public void draw ()
{
System.out.println(this.getClass().getName()+"draw()"
+this.name);
}
}
```

Constructor

Method

```java
//1
public class Circle {
String name;
double x,y,r;

public Circle (String st) {
name = st;
}
public void draw ()
{
System.out.println(this.getClass().getName()+"draw()"
+this.name);
}
}
```

Constructor

Method

What is getClass method returns?
The Java Object getClass() method **returns the class name of the object**. The syntax of the getClass() method is:
object.getClass()

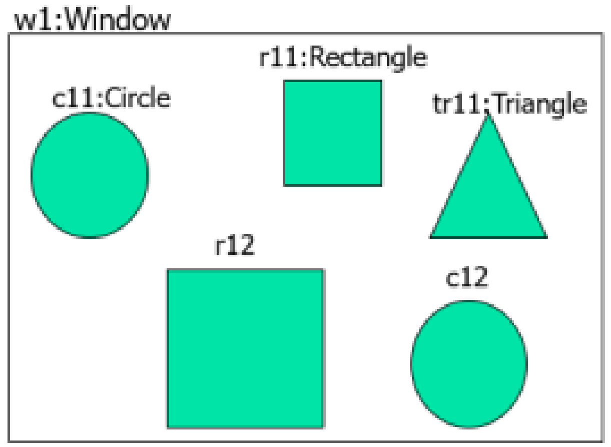
What does getName () do in Java?
getName. **Returns the name of the entity** (class, interface, array class, primitive type, or void) represented by this Class object, **as a String** .

Αρα αν έχω ένα αντικείμενο της Circle που το δημιούργησα ως εξής
new Circle("c21")
Τότε
το αποτέλεσμα της draw() θα είναι
Circledraw()c21

```
//1
public class Circle {
String name;
double x,y,r;

public Circle (String st) {
name = st;
}
public void draw ()
{
System.out.println(this.getClass().getName()+"draw()"
+this.name);
}
}
```

```
//1
public class Rectangle {
String name;
double x1,y1,lenght,width;

public Rectangle (String st) {


}
public void draw ()
{



}

}
```

```
//1
public class Triangle {
String name;








}
}
```

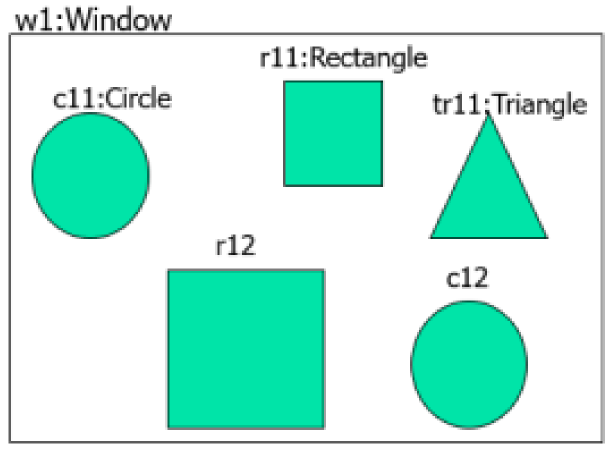


Χρειάζομαι μια κλάση Window

Τι ακριβώς θα κάνει;

```
public class Window {
String title;
          circles;
                rectangles;
             triangles;
```

Θα δημιουργήσω μια Κλάση Window
Τι θα έχω μέσα στο παράθυρο Window ;
Στην πραγματικότητα θα εμφανίζει μηνύματα για κάθε σχήμα που δημιουργεί. ΔΕΝ ΘΑ ΖΩΓΡΑΦΙΖΕΙ ΤΑ ΣΧΗΜΑΤΑ

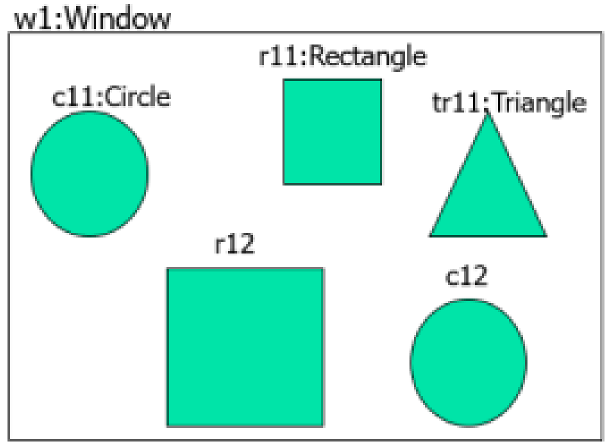


Πως θα έχω παραπάνω από ένα circle , ένα rectangle και ένα triangle;

```
public class Window {
String title;
Circle[] circles;
Rectangle[] rectangles;
Triangle[] triangles;




}
}
```

ΘΑ δημιουργήσω **ARRAYS** από circles,
rectangles και triangles
Και πως αυτά θα δημιουργούνται στο Window;

```
public class Window {
String title;
Circle[] circles;
Rectangle[] rectangles;
Triangle[] triangles;
int circlesIndex=0;
int rectanglesIndex=0;
int trianglesIndex=0;

public Window(String t)
{
this.title= t;
circles = new Circle[100];
rectangles = new …………….;
triangles = new ……………….;

}
}
```



Και πως προσθέτω circles, rectangles και triangles;
Μήπως χρειάζομαι μεθόδους για να το κάνω αυτό;

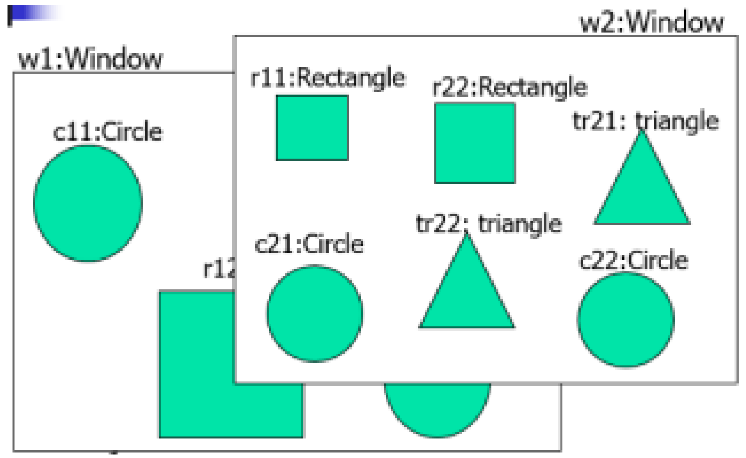
```
public class Window {
String title;
Circle[] circles;
Rectangle[] rectangles;
Triangle[] triangles;
int circlesIndex=0;
int rectanglesIndex=0;
int trianglesIndex=0;

public Window(String t)
{
this.title= t;
circles = new Circle[100];
rectangles = new                          ;
triangles = new

}
public void draw()
{
System.out.println(this.getClass().getName()+".draw()"+this.title);
}
```

```
public void add(Circle c) {
circles[circlesIndex++]=c;
c.draw();
}
public void add(Rectangle r) {


}
public void add(Triangle t) {


}
}
```

ΙΔΙΟ ΟΝΟΜΑ ΜΕΘΟΔΟΥ ΔΙΑΦΟΡΕΤΙΚΕΣ παράμετροι
Overloading

Θέλω να μπορώ να φέρω το παράθυρο toFront .
Αυτό είναι πιο κατανοητό αν έχω δημιουργήσει δυο στιγμιότυπα της Window το w1 και w2.
Ποιο από τα δύο έχει έρθει “toFront”

Αρα χρειάζομαι μια ακόμα μέθοδο BringToFront



```
public class Window {
String title;
Circle[] circles;
Rectangle[] rectangles;
Triangle[] triangles;
int circlesIndex=0;
int rectanglesIndex=0;
int trianglesIndex=0;

public Window(String t)
{
this.title= t;
circles = new Circle[100];
rectangles = new
triangles = new

}
public void draw()
{
System.out.println(this.getClass().getName()+".draw()"
+this.title);
}
```

```
public void bringToFront() {
System.out.println(this.getClass().getName()+".bringToFront()
"+this.title);
draw();
for(int i=0;i<circles.length && circles[i]!=null; i++)
circles[i].draw();
//αντιστοιχα για rectangle και triangle
for


 //αντιστοιχα για triangle
for



}
public void add(Circle c) {
circles[circlesIndex++]=c;
c.draw();
}
public void add(Rectangle r) {



}
public void add(Triangle t) {
```

```
//1
import java.lang.*;
public class ShapesHandlingApp {
public static void main(String[] args) {
// TODO Auto-generated method stub
Window w1 = new Window("w1");
w1.draw();
w1.add(new Circle("c11"));
w1.add(new Rectangle("r11"));
//δημιουργηστε και προσθεστε τα στιγμιοτυπα
Ractangle και Triangle


//Δημιουργηστε νέο παραθυρο
Window w2= new Window("w2");
w2.draw();
w2.add(new Circle("c21"));
//ομοιως προσθεστε rectanlge

w1.bringToFront();
}
}
```

```
public class Window {
String title;
Circle[] circles;
Rectangle[] rectangles;
Triangle[] triangles;
int circlesIndex=0;
int rectanglesIndex=0;
int trianglesIndex=0;

public Window(String t)
{
this.title= t;
circles = new Circle[100];
rectangles =
triangles =

}
public void draw()
{
System.out.println(this.getClass().get
Name()+".draw()"+this.title);
}
```

```
//1
public class Circle {
String name;
double x,y,r;

public Circle (String st) {
name = st;
}
public void draw ()
{
System.out.println(this.getClass()
.getName()+"draw()"+this.name);
}
}
```

```java
//1
import java.lang.*;
public class ShapesHandlingApp {
public static void main(String[]
args) {
// TODO Auto-generated method stub
Window w1 = new Window("w1");
w1.draw();
w1.add(new Circle("c11"));
w1.add(new Rectangle("r11"));
w1.add(new Circle("c12"));
w1.add(new Rectangle("r12"));
w1.add(new Triangle("tr11"));

Window w2= new Window("w2");
w2.draw();
w2.add(new Circle("c21"));
w2.add(new Rectangle("r21"));
w1.bringToFront();
}
}
```

```java
public class Window {
String title;
Circle[] circles;
Rectangle[] rectangles;
Triangle[] triangles;
int circlesIndex=0;
int rectanglesIndex=0;
int trianglesIndex=0;

public Window(String t)
{
this.title= t;
circles = new Circle[100];
rectangles = new Rectangle[100];
triangles = new Triangle[100];

}




}
```

Πόσα circles, triangles, rectangles μπορώ να έχω;

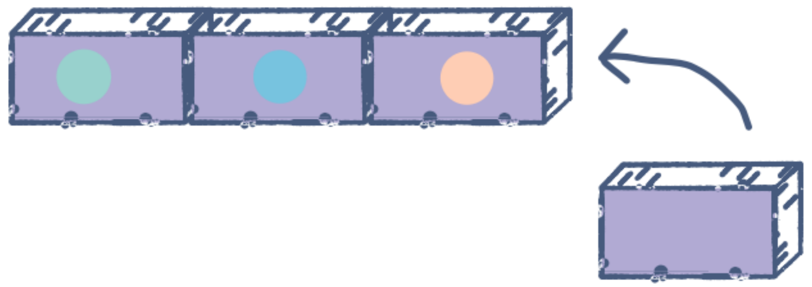
Χρήση κλάσης ArrayList

The `ArrayList` class is a resizable array, which can be found in the `java.util` package.

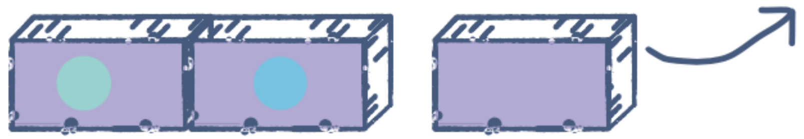
The difference between a built-in array and
an `ArrayList` in Java, is that
the size of an array cannot be modified.
While elements can be added and removed from
an `ArrayList` whenever you want.

While built-in arrays have a fixed size, ArrayLists can *change* their size dynamically.

Elements can be added and removed from an ArrayList whenever there is a need, helping the user with memory management.



Adding a new box when all other boxes are filled



Removing a box when it is empty and taking up extra memory space

- Specify the data type of elements in an ArrayList
- Import the ArrayList class

```
import java.util.ArrayList; //import the ArrayList class
class MyClass {
    public static void main( String args[] ) {
    ArrayList<String> shapes = new ArrayList<String>(); // Create an ArrayList object with a string data type
    }
}
```

Χρήσιμες μέθοδοι

Add an item: The add() method is used to add an item at the start of an ArrayList. The index of this item is 0 and all other indexes are increased by 1.
shapes.**add("**hexagon")

Access an item: The get() method, taking an index as input, is used to access an element in the ArrayList.
shapes.**get**(3)

Set an item: The set() method, taking an index as input, is used to set an element in the ArrayList at the specified index.
shapes.**set**(2, "triangle")

Remove an item: The remove() method, taking an index as input, is used to remove an element in an ArrayList. Indexes of all the elements in front of the removed element are reduced by 1.
shapes.**remove**(1)

Remove all items: The clear() method is used to remove all elements in an ArrayList.
shapes.**clear**()

Size of ArrayList: The size() method is used to find the number of elements in an ArrayList.
shapes.**size**()

Οι κλάσεις Circle, Rectangle και Triangle θα αλλάξουν;

```
//1
public class Circle {
String name;
double x,y,r;

public Circle (String st) {
name = st;
}
public void draw ()
{
System.out.println(this.getClass().getName()+"draw()"+this.name);
}
}
```

```
//1
public class Rectangle {
String name;
double x1,y1,lenght,width;

public Rectangle (String st) {


}
public void draw ()
{



}

}
```

```
//1
public class Triangle {
String name;
public Triangle (String st) {

}
public void draw ()
{


}
}
```

```java
//ArrayList
import java.util.ArrayList;
public class Window {
String title;
ArrayList <Circle> circles;
ArrayList <Rectangle> rectangles;
ArrayList <Triangle> triangles;
//int circlesIndex=0;
//int rectanglesIndex=0;
//int trianglesIndex=0;

public Window(String t)
{
this.title= t;

circles = new ArrayList<Circle> ();
rectangles= new ArrayList <Rectangle> ();
triangles = new ArrayList <Triangle>();

}

public void draw()
{
System.out.println(this.getClass().getName()+".draw
()"+this.title);
}
```

```java
//ArrayList
import java.util.ArrayList;
public class Window {
String title;
ArrayList <Circle> circles;
ArrayList <Rectangle> rectangles;
ArrayList <Triangle> triangles;
//int circlesIndex=0;
//int rectanglesIndex=0;
//int trianglesIndex=0;

public Window(String t)
{
this.title= t;

circles = new ArrayList<Circle> ();
rectangles= new ArrayList <Rectangle> ();
triangles = new ArrayList <Triangle>();

}

public void draw()
{
System.out.println(this.getClass().getName()+"I draw() the window "+this.title); }
```

```java
public void bringToFront() {

System.out.println(this.getClass().getName()+".bringToFront()"+this.title);
draw();
      int sizec=circles.size();



      for (int i =0; i<sizec; i++ )
      {circles.get(i).draw();}
//ομοιως για rectangles, triangles


}
public void add(Circle c) {
circles.add(c);
//[circlesIndex++]=c;
c.draw();
}
public void add(Rectangle r) {



}
public void add(Triangle t) {



}}
```

```
//ArrayList
import java.util.ArrayList;
import java.lang.*;
public class ShapesHandlingApp {

public static void main(String[] args) {

Window w1 = new Window("w1AL");
Rectangle r11= new Rectangle("r11AL ");
w1.draw();
w1.add(new Circle("c11AL "));
w1.add(r11);
w1.add(new Circle("c12AL "));
w1.add(new Rectangle("r12AL "));
w1.add(new Triangle("tr11AL "));

Window w2= new Window("w2AL ");
w2.draw();
w2.add(new Circle("c21AL "));
w2.add(new Rectangle("r21AL "));
w1.bringToFront();
            }

}
```

Όταν εκτελεστεί το πρόγραμμά μου εμφανίζεται :

```
WindowI draw() the window w1AL
Circledraw()  c11AL
Rectangledraw()  r11AL
Circledraw()  c12AL
Rectangledraw()  r12AL
Triangledraw()  tr11AL
WindowI draw() the window w2AL
Circledraw()  c21AL
Rectangledraw()  r21AL
Window.bringToFront() w1AL
WindowI draw() the window w1AL
size of ArryList Circles is  2
size of ArryList Rectangles is  2
Circledraw()  c11AL
Circledraw()  c12AL
Rectangledraw()  r11AL
Rectangledraw()  r12AL
Triangledraw()  tr11AL
```

Με Array, ArrayList και Vectors…

The `ArrayList` class is a resizable array, which can be found in the `java.util` package.

The difference between a built-in array and an `ArrayList` in Java, is that the size of an array cannot be modified. While elements can be added and removed from an `ArrayList` whenever you want.

Vector is like the dynamic array which can grow or shrink its size. Unlike array, we can store n-number of elements in it as there is no size limit.

```
//ARRAY_ARRYLIST_VECTOR
import java.util.ArrayList;
import java.util.Vector;
public class Window {
        String title;

        ArrayList<Circle> circles;

        Vector<Rectangle> rectangles;

        Triangle[] triangles;
        int trianglesIndex=0;

        public Window(String t)
        {
                this.title= t;

                circles = new ArrayList<Circle>();

                rectangles = new Vector<Rectangle>();

                triangles = new Triangle[100];
        }

        public void draw()
        {

        System.out.println(this.getClass().getName()+".draw()"+this.title);
        }
```

```
public void bringToFront() {

        System.out.println(this.getClass().getName()+".bringToFront()"+this.title);
                draw();
// δημιουργια επαναληψεων για την εκτελεση της draw() για circles, rectangles,
triangles
                for(Circle c:circles)
                        c.draw();

                for (int i=0; i<rectangles.size();i++)

                for (Triangle t:triangles) {


                        }
        }
// μεθοδοι add() για circles, rectangles, triangles

        public void add(Circle c) {
                circles.add(c);
                c.draw();
        }
        public void add(Rectangle r) {



        }
        public void add(Triangle t) {



```

Στις διαφάνειες παρουσιάζονται τμήματα κώδικα και αποτελέσματα από την εκτέλεση.

Ο κώδικας που αποκρύπτεται , έχει παρουσιαστεί και επεξηγηθεί στη διάλεξη του φροντιστηρίου αναλυτικά