

Αντικειμενοστρεφής Προγραμματισμός (Object-Oriented Programming)

(CEID_NNY106)

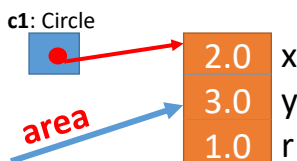
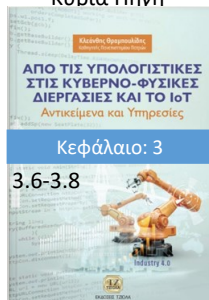
Java as an OOP Language

Java

High-level programming
language

Part A

Κύρια Πινή



Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

[Wikipedia](#)

Designed by: [James Gosling](#)

First appeared: May 23, 1995; 27 years ago

Paradigm: Multi-paradigm: generic, object-oriented (class-based), functional, imperative, reflective, concurrent

Kleanthis Thramboulidis

Prof. of Software and System Engineering

University of Patras

<https://sites.google.com/site/thramboulidiskleanthis/>

Οργάνωση Διάλεξης

- **Κλάση, Στιγμιότυπο, Μεθοδος**
 - **Ορισμός Κλάσης**
 - Η μέθοδος main()
 - **Δημιουργία Στιγμιότυπου**
 - **Αναφορά σε μεταβλητές στιγμιότυπου**
 - **Αποστολή μηνύματος σε στιγμιότυπο**
- **Δημιουργός (Constructor)**
 - Ο τελεστής this
- **Data Encapsulation - Accessors και Mutators**

Η κλάση (class) και η μέθοδος main()

```

/*****

```

- * File name : **HelloWorld.java**
- * Compile with javac: javac HelloWorld.java
- * Execute with java: java HelloWorld
- *
- * *Developer's Comments*
- * The HelloWorld class implements an application that simply
- * displays "Hello World!" to the standard output.

```

*****/

```

```

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}

```

Υλοποιώ την συμπεριφορά του προγράμματος με την μέθοδο main()

Δραστηριότητα 3.2 Το πρόγραμμα HelloWorld

Αναπτύξτε ένα πρόγραμμα σε Java που θα τυπώνει στην οθόνη τη φράση «Hello World!».

Αναπαριστώ το πρόγραμμα ως μια κλάση

```

8 public class HelloWorld{
9     static String message = "Hello World!";
10
11     public static void main(String [] args){
12         System.out.println(message);
13     }

```

© 2023 Κλεάνθης Θραμπουλίδης

Java

Η Άσκηση – Circles Handling App

Δραστηριότητα 4.1 Circles Handling App

Γράψτε τον πηγαίο κώδικα ενός προγράμματος το οποίο **θα δημιουργεί δύο κύκλους** με δεδομένα χαρακτηριστικά και θα τυπώνει για τον καθένα την επιφάνεια και την περίμετρό του.

στιγμιότυπα της κλάσης **Circle**

κλάση **CirclesHandlingApp**

© 2023 Κλεάνθης Θραμπουλίδης

Java as an OOP language Part 1

Διαφάνεια 4

Το πρόγραμμα Circles Handling App ως κλάση

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named 'Blue: CirclesHandlingApp V0' with a class named 'CirclesHandlingApp'.
- Code Editor:** Displays the source code for 'CirclesHandlingApp.java':


```

1  /**
2  * Write a description of class CirclesHandlingApp here.
3  *
4  * @author (your name)
5  * @version (a version number or a date)
6  */
7
8  public class CirclesHandlingApp{
9
10     public static void main(String []args){
11         System.out.println("CirclesHandlingApp running");
12     }
13 }
      
```
- Terminal Window:** Shows the output 'CirclesHandlingApp running'.

© 2023 Κλεάνθης Θραμπουζίδης Java as an OOP language Part 1 Διαφάνεια 5

Λεκτική περιγραφή της main()

```

8  public class CirclesHandlingApp{
9
10     public static void main(String []args){
11         System.out.println("CirclesHandlingApp running");
12         // Δημιουργησε κύκλο με κέντρο 2.0, 3.0 και ακτίνα 1.0
13         // Δημιουργησε κύκλο με κέντρο 2.0, 3.0 και ακτίνα 2.0
14         // Τύπωσε κέντρο και ακτίνα και την επιφάνεια και την
15         // περίμετρο του κύκλου (2.0, 3.0, 1.0)
16         // Τύπωσε κέντρο και ακτίνα και την επιφάνεια και την
17         // περίμετρο του κύκλου (2.0, 3.0, 2.0)
18     }
19 }
      
```

Ορισμός κλάσης Circle

```
public class Circle {
    // data members
    public double x,y;
    public double r;
    // methods
    public double circumference() {
        return 2*3.14*r;
    }
    public double area() {
        return 3.14*r*r;
    }
}
```

Ορίζουν τη **δομή** των στιγμιότυπων της κλάσης

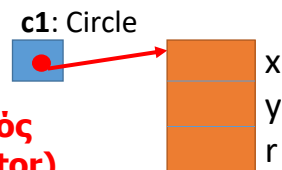
Ορίζουν τη συμπεριφορά των στιγμιότυπων της κλάσης

Δημιουργία στιγμιότυπου (Instance)

- Ορισμός αναφοράς (reference)

Circle c1; //c1 είναι απλά ένας identifier

Δημιουργός (constructor)



- Δημιουργία στιγμιότυπου και ανάθεση στην αναφορά

c1 = new Circle(); // όλα τα αντικείμενα πρέπει να δημιουργούνται δυναμικά

- Εναλλακτικά**

Circle c1 = new Circle();

var c1 = new Circle();

υπάρχει ένας τουλάχιστο **constructor** που έχει το όνομα της κλάσης και εκτελεί την απαραίτητη αρχικοποίηση για το νέο αντικείμενο

Αναφορά σε μεταβλητές στιγμιότυπου

```
Circle c1 = new Circle();
var c2 = new Circle();
```

- αρχικοποίηση δεδομένων στιγμιότυπου

```
// αρχικοποίηση των στοιχείων του στιγμιότυπου c1
```

```
c1.x = 2.0;
```

```
c1.y = 3.0;
```

```
c1.r = 1.0;
```

```
// αρχικοποίηση των στοιχείων του στιγμιότυπου c2
```

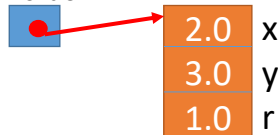
```
c2.x = 2.0;
```

```
c2.y = 3.0;
```

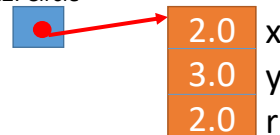
```
c2.r = 2.0;
```

```
21 System.out.println("c1.x= " + c1.x + "\tc1.y= " + c1.y + "\tc1.r= " + c1.r );
22 System.out.println("c2.x= " + c2.x + "\tc2.y= " + c2.y + "\tc2.r= " + c2.r );
```

c1: Circle



c2: Circle



Blue: Terminal Window - CirclesHandlingApp

Options

```
c1.x= 2.0      c1.y= 3.0      c1.r= 1.0
c2.x= 2.0      c2.y= 3.0      c2.r= 2.0
```

CirclesHandlingApp V1.0

```
8 public class CirclesHandlingApp{
9     static Circle c1,c2;
10
11     public static void main(String []args){
12         c1= new Circle();
13         c2= new Circle();
14         c1.x = 2.0;
15         c1.y=3.0;
16         c1.r=1.0;
17         c2.x = 2.0;
18         c2.y=3.0;
19         c2.r=2.0;
20         System.out.println("c1.x= " + c1.x + "\tc1.y= " + c1.y + "\tc1.r= " + c1.r );
21         System.out.println("c2.x= " + c2.x + "\tc2.y= " + c2.y + "\tc2.r= " + c2.r );
22     }
23 }
```

```
8 public class Circle{
9     double x,y,r;
10
11     public double circumference() {
12         return 2*3.14*this.r;
13     }
14
15     public double area() {
16         return 3.14*this.r*this.r;
17     }
18 }
```

Blue: CirclesHandlingApp V1.0

object Edit Tools View Help

View Class...

→

Compile

CirclesHandlingApp

Circle

Blue: Terminal Window - CirclesHandlingApp

Options

```
c1.x= 2.0      c1.y= 3.0      c1.r= 1.0
c2.x= 2.0      c2.y= 3.0      c2.r= 2.0
```

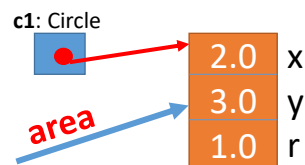
Αποστολή μηνύματος – Κλήση μεθόδου

```
Circle c = new Circle();
```

- κλήση μεθόδου

```
double a;
```

```
a = c.area();
```



```
a = area(c);
```

procedural programming

CirclesHandlingApp V1.1

```

8 public class CirclesHandlingApp{
9     static Circle c1,c2;
10
11     public static void main(String []args){
12         c1= new Circle();
13         c2= new Circle();
14         c1.x = 2.0;    c1.y=3.0;    c1.r=1.0;
15         c2.x = 2.0;    c2.y=3.0;    c2.r=2.0;
16         System.out.println("c1.x= " + c1.x + "\tc1.y= " + c1.y +"\tc1.r= " + c1.r );
17         System.out.println("c1.area= " + c1.area() + "\tc1.circumference = " + c1.circumference());
18         System.out.println("c2.x= " + c2.x + "\tc2.y= " + c2.y +"\tc2.r= " + c2.r );
19         System.out.println("c2.area= " + c2.area() + "\tc2.circumference = " + c2.circumference());
20     }
21 }

```

```

8 public class Circle{
9     double x,y,r;
10
11     public double circumference() {
12         return 2*3.14*this.r;
13     }
14
15     public double area() {
16         return 3.14*this.r*this.r;
17     }
18 }

```

BlueJ: Terminal Window - CirclesHandlingApp V1.1

Options

```

c1.x= 2.0      c1.y= 3.0      c1.r= 1.0
c1.area= 3.14  c1.circumference = 6.28
c2.x= 2.0      c2.y= 3.0      c2.r= 2.0
c2.area= 12.56 c2.circumference = 12.56

```

Το πρόγραμμα Echo

```

9 public class Echo
10 {
11     public static void main(String[] args){
12         var sc = new Scanner(System.in);
13         String message;
14
15         System.out.println("Echo running - write \"bye\" to
16         while(true){
17             message = sc.nextLine();
18             System.out.println(message);
19             if(message.equals("bye") ){
20                 break;
21             }
22         }
23         sc.close();

```

message = sc.nextLine();

message.equals("bye")

sc.close();

Η κλάση Scanner

Overview Package **Class** Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

java.util

Class Scanner

java.lang.Object
java.util.Scanner

All Implemented Interfaces:
Closeable, AutoCloseable, Iterator<String>

public final class **Scanner**
extends Object
implements Iterator<String>, Closeable

A simple text scanner which can parse primitive types and strings using regular expressions.

A Scanner breaks its input into tokens using a delimiter pattern, which by default matches whitespace. The resulting tokens may then be converted into values of different types using the various next methods.

For example, this code allows a user to read a number from System.in:

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

Δημιούργησε στο BlueJ στιγμιότυπο της Scanner και δοκίμασε να του στείλεις μηνύματα

Οργάνωση Διάλεξης

- Κλάση και Στιγμιότυπο (Class and Instance)
 - Ορισμός Κλάσης
 - Δημιουργία Στιγμιότυπου
 - Αναφορά σε μεταβλητές στιγμιότυπου
 - Αποστολή μηνύματος σε στιγμιότυπο
- **Δημιουργός (Constructor)**
 - Ο τελεστής **this**
- Data Encapsulation - Accessors και Mutators

Ορισμός Δημιουργού (constructor)

```
public class Circle {
  public double x, y;
  public double r;
  public Circle(double x1, double y1, double r1){
    x=x1;
    y=y1;
    r=r1;
  }
}
```

Ο Δημιουργός έχει πάντα το όνομα της Κλάσης

Δημιουργός – Τελεστής this

- Constructor κλάσης Circle

```
public Circle(double x, double y, double r) {
    this.x=x;
    this.y=y; ?
    this.r=r;
}
```

Κανόνες

- το όνομα του constructor είναι πάντα ίδιο με το όνομα της κλάσης
- δεν ορίζεται επιστρεφόμενη τιμή
- δεν χρησιμοποιείται το return
- Ο δημιουργός επιστρέφει έμμεσα τον τελεστή **this**

Ο τελεστής this

Book: Ενότητα 5.2.1

- **αναφέρεται**
 - στο αντικείμενο στο οποίο γίνεται αναφορά
- **χρησιμοποιείται**
 - από τον δημιουργό για την αναφορά στα μέλη του στιγμιότυπου που μόλις δημιουργήθηκε
 - από τις μεθόδους για αναφορά στα δεδομένα του αντικειμένου πάνω στο οποίο αυτές δρουν
 - για να κάνει τον κώδικα περισσότερο διαυγή
- **απαραίτητος όταν** η μέθοδος έχει
 - τοπική μεταβλητή με το ίδιο όνομα ή
 - τυπικό όρισμα με το ίδιο όνομα

```
public Circle(double x, double y
    x=x;
    y=y;
    r=r;
}
```

```
public double area() {
    return 3.14*this.r*this.r;
}
```

```
public boolean hasRadius(double r){
    return (r==this.r)? true: false;
}
```

CirclesHandlingApp V2.0

```

8 public class CirclesHandlingApp{
9     static Circle c1,c2;
10
11     public static void main(String []args){
12         c1 = new Circle(2.0,3.0,1.0);
13         c2 = new Circle(2.0,3.0,2.0);
14         // c1.x = 2.0;
15         // c1.y=3.0;
16         // c1.r=1.0;
17         // c2.x = 2.0;
18         // c2.y=3.0;
19         // c2.r=2.0;
20         System.out.println("c1.x= " + c1.x + "\tc1.y= " +
21         System.out.println("c2.x= " + c2.x + "\tc2.y= " +
22         System.out.println(c1.hasRadius(1.0));
23         System.out.println(c1.hasRadius(2.0));
24     }
25 }

```

```

8 public class Circle{
9     double x,y,r;
10
11     public Circle(double x, double y, double r){
12         this.x=x;
13         this.y=y;
14         this.r=r;
15     }
16     public double circumference() {
17         return 2*3.14*this.r;
18     }
19     public double area() {
20         return 3.14*this.r*this.r;
21     }
22
23     public boolean hasRadius(double r){
24         //return (r==this.r)? true: false;
25         return (r==this.r);
26     }
27 }

```

BlueJ: Terminal Window - CirclesHandlingApp

Options		
c1.x= 2.0	c1.y= 3.0	c1.r= 1.0
c2.x= 2.0	c2.y= 3.0	c2.r= 2.0
true		
false		

© 2023 Κλεάνθης Θραμπουλιδης Java as an OOP language Part 1 Διαφάνεια 19

CirclesHandlingApp V3.0

```

8 public class CirclesHandlingApp
9 {
10     static Circle c1,c2;
11
12     public static void main(String []args){
13         c1 = new Circle(2.0,3.0,1.0);
14         c2 = new Circle(2.0,3.0,2.0);
15         // c1.x = 2.0;
16         // c1.y=3.0;
17         // c1.r=1.0;
18         // c2.x = 2.0;
19         // c2.y=3.0;
20         // c2.r=2.0;
21         // System.out.println("c1.x= " + c1.x + "\t" +
22         // System.out.println("c2.x= " + c2.x + "\t" +
23         c1.print();
24         c2.print();
25         System.out.println(c1.hasRadius(1.0));
26         System.out.println(c1.hasRadius(2.0));
27     }
28 }

```

Δείτε εναλλακτικά την System.out.println(c1);

```

8 public class Circle{
9     private double x,y,r;
10
11     public Circle(double x, double y, double r){
12         this.x=x;
13         this.y=y;
14         this.r=r;
15     }
16     public double circumference() {
17         return 2*3.14*this.r;
18     }
19     public double area() {
20         return 3.14*this.r*this.r;
21     }
22
23     public boolean hasRadius(double r){
24         return (r==this.r);
25     }
26
27     public void print(){
28         System.out.println("x= " + this.x + "\ty= " +
29         + this.y + "\tr= " + this.r );
30     }
31 }

```

BlueJ: Terminal Window - CirclesHandlingApp

Options		
x= 2.0	y= 3.0	r= 1.0
x= 2.0	y= 3.0	r= 2.0
true		
false		

© 2023 Κλεάνθης Θραμπουλιδης Java as an OOP language Part 1 Διαφάνεια 20

Οργάνωση Διάλεξης

- Κλάση και Στιγμιότυπο (Class and Instance)
 - Ορισμός Κλάσης
 - Δημιουργία Στιγμιότυπου
 - Αναφορά σε μεταβλητές στιγμιότυπου
 - Αποστολή μηνύματος σε στιγμιότυπο
- Δημιουργός (Constructor)
 - Ο τελεστής this
- **Data Encapsulation - Accessors και Mutators**

Data encapsulation - Accessors and Mutators

- One of the ways we can **enforce data encapsulation** is through the use of accessors and mutators. The role of accessors and mutators are to return and set the values of an object's state.

Accessor Methods

- An accessor method is used to return the value of a private field. It follows a naming scheme prefixing the word "**get**" to the start of the method name.

```
// Accessor for firstName
public String getFirstName() {
    return firstName;
}
```

Mutator Methods

- A mutator method is used to set a value of a private field. It follows a naming scheme prefixing the word "**set**" to the start of the method name.

```
// Mutator for address
public void setAddress(String address) {
    this.address = address;
}
```

CirclesHandlingApp V4.0

```

8 public class CirclesHandlingApp
9 {
10     static Circle c1,c2;
11
12     public static void main(String []args){
13         c1 = new Circle(2.0,3.0,1.0);
14         c2 = new Circle(2.0,3.0,2.0);
15         c1.print();
16         c2.print();
17         System.out.println(c1.hasRadius(1.0));
18         c1.setR(12.2);
19         c1.print();
20         System.out.println(c1.hasRadius(1.0));
21         System.out.println(c1.hasRadius(12.2));
22         System.out.println(c1.hasRadius(2.0));
23     }
24 }

```

Blue: Terminal Window - CirclesHandlingApp V4.0

Options

```

x= 2.0 y= 3.0 r= 1.0
x= 2.0 y= 3.0 r= 2.0
true
x= 2.0 y= 3.0 r= 12.2
false
true
false

```

```

8 public class Circle
9 {
10     private double x,y,r;
11
12     public Circle(double x, double y, double r){
13         this.x=x;    this.y=y;    this.r=r;
14     }
15     public double circumference() {
16         return 2*3.14*this.r;}
17
18     public double area() {
19         return 3.14*this.r*this.r;}
20
21     public boolean hasRadius(double r){
22         return (r==this.r)? true: false;}
23
24     public void print(){
25         System.out.println("x= " + this.x + "\ty= "
26             + this.y +"\tr= " + this.r );
27     }
28
29     public double getR(){ return this.r;}
30
31     public void setR(double r){ this.r=r;}
32 }

```