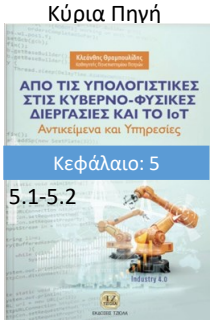
Αντικειμενοστρεφής Προγραμματισμός (Object-Oriented Programming)

(CEID_NNY106)

Java as an OOP Language Part B

Κύρια Πηγή




```
class Ex1c2 {
    public static void main(String[] argv) {
        MyClass myObject = new MyClass();
        myObject.d1 = new Double(12.0);
        myObject.d2= new Double(24.0);
        myObject.d3 = myObject.d1 + myObject.d2;
        System.out.println(myObject.d1 + "+" +
                                myObject.d2 + "=" + myObject.d3);
    }
}

class MyClass {
    Double d1, d2, d3;
    }
```

Kleanthis Thramboulidis
Prof. of Software and System Engineering
University of Patras
https://sites.google.com/site/thramboulidiskleanthis/

Java

High-level programming language

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

Δραστηριότητα 5.1 – Circles Sorting App

Δραστηριότητα 5.1
Circles Sorting App

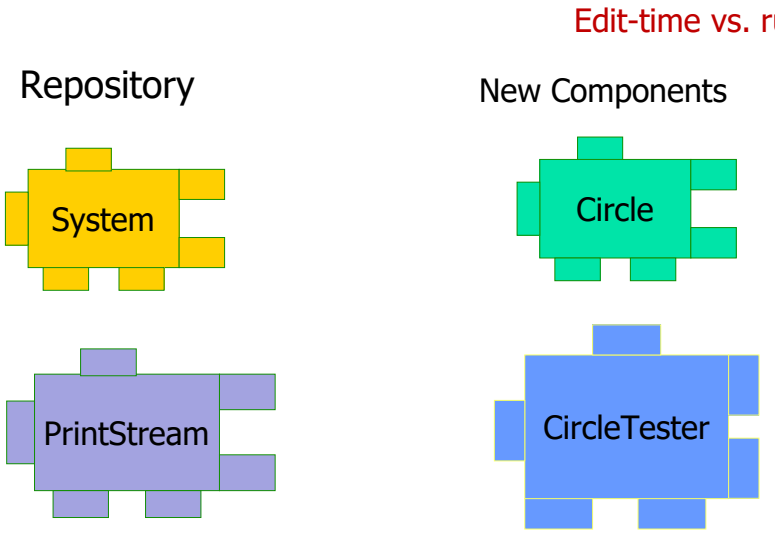
Αναπτύξτε ένα πρόγραμμα το οποίο να δημιουργεί δεδομένο αριθμό από κύκλους με δεδομένα χαρακτηριστικά (hardcoded) και να ικανοποιεί τις παρακάτω απαιτήσεις:

Requirement No 1: Να τυπώνει για τον καθένα την επιφάνεια και την περίμετρό του.

Requirement No 2: Να μπορούμε να ενημερώσουμε τον χρήστη, ανά πάσα στιγμή κατά τη διαδικασία εκτέλεσης του προγράμματος, για τον αριθμό των στιγμιότυπων της κλάσης Circle που έχουν δημιουργηθεί.

Requirement No 3: Να ταξινομεί τους κύκλους και να τυπώνει για τον καθένα την επιφάνεια και την περίμετρό του.

Δραστηριότητα 5.1 – The constituent components



Οργάνωση Διάλεξης

- **Δημιουργία Στιγμιότυπων**
  - **Πολλαπλοί Δημιουργοί**
  - **Υπερφόρτωση μεθόδων (method overloading)**
  - **Keyword this**
- Class data members, σταθερές
- Class methods
  - Factory methods
- Java code style Guidelines
- Ασκήσεις

Πολλαπλοί Δημιουργοί – Υπερφόρτωση Μεθόδων

- ανάγκη απόδοσης αρχικών τιμών σε στιγμιότυπο με διαφορετικούς τρόπους
- εναλλακτικοί Δημιουργοί της κλάσης Circle

```
public Circle(double r)
        {x=0.0;y=0.0;this.r=r;}
public Circle(Circle c)
        {x=c.x;y=c.y;r=c.r;}
public Circle()
        {x=0.0;y=0.0;r=1.0;}
```

default constructor
If there is no default or parameterized constructor written by user, Java automatically creates default constructor.
Default constructor initializes member data variable to default values

Κλήση δημιουργού – keyword this

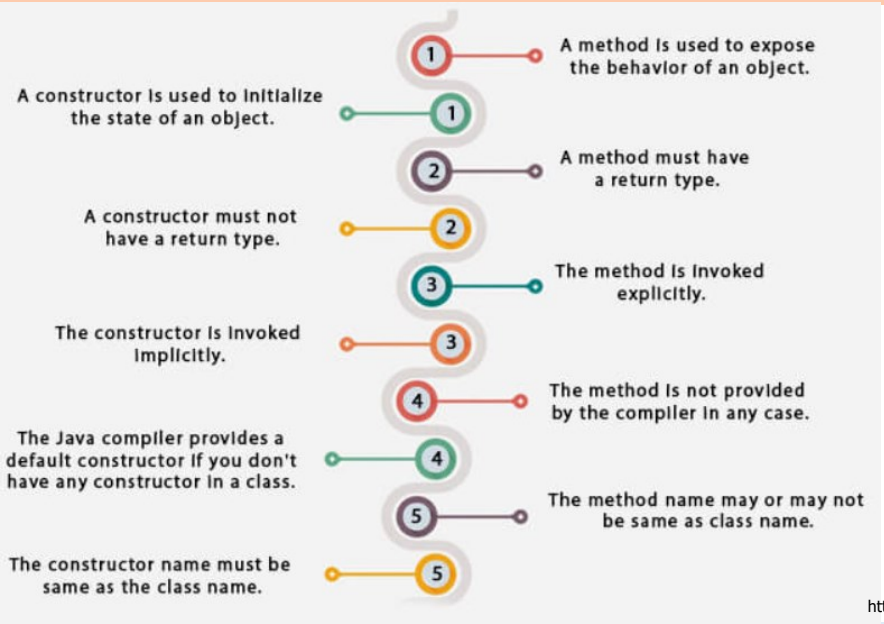
- ένας constructor έχει τη δυνατότητα κλήσης άλλου constructor

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

public Circle(double r) {
                this(0.0,0.0,r);}

public Circle(Circle c) {
                this(c.x,c.y,c.r);}

public Circle() {
                this(0.0,0.0,1.0);}
```

```
public Circle(double r)
        {x=0.0;y=0.0;this.r=r;}
public Circle(Circle c)
        {x=c.x;y=c.y;r=c.r;}
public Circle()
        {x=0.0;y=0.0;r=1.0;}
```

μόνο ως πρώτη πρόταση του νέου constructor

Constructor vs. Method



https://www.javatpoint.com/java-constructor

Οργάνωση Διάλεξης

- Πολλαπλοί Δημιουργοί
  - Keyword this
- **Class data members, σταθερές**
- Class methods
  - Factory methods
- Java code style Guidelines
- Ασκήσεις

class variables – Δήλωση

Requirement No 2

Requirement No 2: Να μπορούμε να ενημερώσουμε τον χρήστη, ανά πάσα στιγμή κατά τη διαδικασία εκτέλεσης του προγράμματος, για τον αριθμό των στιγμιότυπων της κλάσης Circle που έχουν δημιουργηθεί.

Παράδειγμα

```
public class Circle {
  static int numOfInstances = 0;        //class variable
  public double x,y,r;

// methods
public Circle(double x, double y, double r) {
    this.x=x;this.y=y;this.r=r;
     numOfInstances++;
   }
  public double circumference(){return 2*3,14*r;}
  public double area() {return 3.14*r*r;}
 protected void finalize(){
              numOfInstances--;
             }
  }
```

finalize(): Δεσμευμένη λέξη της Java για να αναθέσουμε έργο στον συλλογέα σκουπιδιών (Garbage Collector)

class variables – Τρόπος Αναφοράς

- *από άλλο αντικείμενο*

***Circle*.numOfInstances**

- στον ορισμό της κλάσης

numOfInstances

```
public Circle(double x, double y, double r) {
        this.x;this.y;this.r;
         numOfInstances ++;
        }
```

~~c1.numOfInstances~~

- αντίθετα με τις instance variables υπάρχουν και μπορούν να χρησιμοποιηθούν χωρίς να απαιτείται instantiation της κλάσης
- καλούνται και **static variables**
- αντικαθιστούν τις γενικές μεταβλητές της C χωρίς το πρόβλημα της σύγκρουσης ονομάτων (name conflict)

Σταθερές (Constants)

Χρήση κεφαλαίων όπως στη C

```
public class Circle {
    public static final double PI = 3.1415926535;
    public double x,y,r;                    //instance variables
```

ο προσδιοριστής **final** απαγορεύει εκφράσεις της μορφής

~~**Circle.PI = 2;**~~

- ο compiler προϋπολογίζει τιμές εκφράσεων που περιέχουν τη σταθερά, π.χ.

  *2*Cirlce.PI*r*

- αποτελούν αντικατάστατο της #define της C

Οργάνωση Διάλεξης

- Πολλαπλοί Δημιουργοί
  - Keyword this
- Class data members, σταθερές
- **Class methods**
  - **Factory methods**
- Java code style Guidelines
- Ασκήσεις

Μέθοδος Κλάσης (class method)

Requirement No 3

Σύγκριση στιγμιότυπων τύπου Circle

Requirement No 3: Να ταξινομεί τους κύκλους και να τυπώνει για τον καθένα την επιφάνεια και την περίμετρό του.

```
public class Circle {
   public double x,y,r;
    // ως μέθοδος στιγμιότυπου
   public Circle bigger(Circle c){
        return (c.r>r) ? c : this;
        }
   // ή εναλλακτικά ως μέθοδος κλάσης
   public static Circle bigger(Circle a, Circle b) {
        return (a.r>b.r) ? a : b;
        }
   ;
   }
```

Η μέθοδος Κλάσης προσδιορίζει συμπεριφορά Κλάσης

κλήση μεθόδου στιγμιοτύπου

```
Circle a = new Circle(2.0);
Circle b = new Circle(3.0);
Circle c = a.bigger(b);
```

κλήση μεθόδου κλάσης

```
Circle a = new Circle(2.0);
Circle b = new Circle(3.0);
Circle c = Circle.bigger(a,b);
```


class methods / static methods

- αντικαθιστούν τις γενικές functions της C αποφεύγοντας το πρόβλημα της σύγκρουσης ονομάτων (name conflict)
- δεν σχετίζονται με συγκεκριμένο στιγμιότυπο και άρα δεν μπορούν να αναφερθούν στον τελεστή this όπως οι instance methods
- καλούνται διαμέσου της κλάσης
  - **όνομα-κλάσης**.όνομα-μεθόδου
  - στο εσωτερικό της κλάσης αναφέρονται μόνο με το όνομα τους

static factory methods

Double in Java SE 9

Constructors

Constructor	Description
`Double (double value)`	**Deprecated.** It is rarely appropriate to use this constructor. The static factory **`valueOf(double)`** is generally a better choice, as it is likely to yield significantly better space and time performance.
`Double(String s)`	**Deprecated.** It is rarely appropriate to use this constructor. Use **`parseDouble(String)`** to convert a string to a double primitive, or use **`valueOf(String)`** to convert a string to a `Double` object.

- public static Double **valueOf**(double d)
- public static Double **valueOf**(String s)
- public static double **parseDouble**(String s)

Factory Method is a creational design pattern that provides an interface for creating objects in a superclass but allows subclasses to alter the type of objects that will be created.

Οργάνωση Διάλεξης

- Πολλαπλοί Δημιουργοί
  - Keyword this
- Class data members, σταθερές
- Class methods
  - Factory methods
- **Java code style Guidelines**
- Ασκήσεις

Java Code Style Guidelines

- **Class and Interface** names should be **capitalized**.
- Data members should be private
- All alphabetic characters in **static final data member** identifiers should be **upper-case**
- **Data member and local variable** identifiers should start with lower-case alphabetic characters.
- All **method names** should start with lower-case alphabetic characters
- Do not wrap lines
- ….
- *See Java Code Style Guidelines*
  - *https://www.cs.cornell.edu/courses/JavaAndDS/JavaStyle.html*
  - *https://google.github.io/styleguide/javaguide.html*
  - *https://www.oracle.com/technetwork/java/codeconventions-150003.pdf*

Οργάνωση Διάλεξης

- Πολλαπλοί Δημιουργοί
  - Keyword this
- Class data members, σταθερές
- Class methods
  - Factory methods
- Java code style Guidelines
- **Ασκήσεις**
  - **Πρωτογενής τύπος double**
  - **Τύπος Double**

Άσκηση 1 – Πρωτογενής τύπος double

- Αναπτύξτε ένα πρόγραμμα σε Java σύμφωνα με το οποίο το σύστημα θα προσθέτει τους αριθμούς 12.0 και 24.0 και θα δίνει στην κύρια έξοδο το αποτέλεσμα της άθροισης.
- Διαμορφώστε τον πηγαίο κώδικα έτσι ώστε το σύστημα να εκτελεί τις παρακάτω ενέργειες:
  1. Να ορίζει μια μεταβλητή τύπου double και να της αναθέτει αρχική τιμή 12.0.
  2. Να ορίζει μια δεύτερη μεταβλητή τύπου double και να της αναθέτει αρχική τιμή 24.0.
  3. Να υπολογίζει το άθροισμα των δύο παραπάνω και να εκχωρεί το αποτέλεσμα σε μια τρίτη μεταβλητή.
  4. Να τυπώνει την τιμή της τρίτης μεταβλητής (αποτέλεσμα της άθροισης).
- **Δώστε εναλλακτικές υλοποιήσεις**

Άσκηση 1 – Έκδοση 1

1. Να ορίζει μια μεταβλητή τύπου `double` και να της αναθέτει αρχική τιμή 12.0.
2. Να ορίζει μια δεύτερη μεταβλητή τύπου `double` και να της αναθέτει αρχική τιμή 24.0.
3. Να υπολογίζει το άθροισμα των δύο παραπάνω και να εκχωρεί το αποτέλεσμα σε μια τρίτη μεταβλητή.
4. Να τυπώνει την τιμή της τρίτης μεταβλητής (αποτέλεσμα της άθροισης).

```
class Ex1a {
   public static void main(String[] argv) {
        double d1 = 12.0;
        double d2 = 24.0;
        double d3 = d1 + d2;
        System.out.println(d1 + "+" + d2 + "=" + d3);
        }
   }
```

οι τρεις μεταβλητές είναι **τοπικές μεταβλητές μεθόδου**

Άσκηση 1 – Έκδοση 2

```
class Ex1b {
   static double d1, d2, d3;
   public static void main(String[] argv) {
        d1 = 12.0;
        d2 = 24.0;
        d3 = d1 + d2;
        System.out.println(d1 + "+" + d2 + "=" + d3);
        }
   }
```

οι τρεις μεταβλητές είναι **μεταβλητές κλάσης**

Άσκηση 1 – Έκδοση 3a

```
class Ex1c {
   double d1, d2, d3;
   public static void main(String[] argv) {
        Ex1c myObject = new Ex1c();
        myObject.d1 = 12.0;
        myObject.d2= 24.0;
        myObject.d3 = myObject.d1 + myObject.d2;
        System.out.println(myObject.d1 + "+" +
                           myObject.d2 + "=" + myObject.d3);
        }
   }
```

οι τρεις μεταβλητές είναι **μεταβλητές στιγμιότυπου**

Άσκηση 1 – Έκδοση 3b

```
class Ex1c2 {
   public static void main(String[] argv) {
        MyClass myObject = new MyClass();
        myObject.d1 = 12.0;        myObject.d2= 24.0;
        myObject.d3 = myObject.d1 + myObject.d2;
        System.out.println(myObject.d1 + "+" +
                           myObject.d2 + "=" +
   myObject.d3);
        }
   }
```

οι τρεις μεταβλητές είναι **μεταβλητές στιγμιότυπου**

```
class MyClass {
   double d1, d2, d3;
   }
```


Άσκηση 2 – τύπος Double

- Αναπτύξτε ένα πρόγραμμα σε Java σύμφωνα με το οποίο το σύστημα θα προσθέτει τους αριθμούς 12.0 και 24.0 και θα δίνει στην κύρια έξοδο το αποτέλεσμα της άθροισης.
- Διαμορφώστε τον πηγαίο κώδικα έτσι ώστε το σύστημα να εκτελεί τις παρακάτω ενέργειες:
  1. Να ορίζει μια μεταβλητή τύπου **`Double`** και να της αναθέτει αρχική τιμή 12.0.
  2. Να ορίζει μια δεύτερη μεταβλητή τύπου **`Double`** και να της αναθέτει αρχική τιμή 24.0.
  3. Να υπολογίζει το άθροισμα των δύο παραπάνω και να εκχωρεί το αποτέλεσμα σε μια τρίτη μεταβλητή.
  4. Να τυπώνει την τιμή της τρίτης μεταβλητής (αποτέλεσμα της άθροισης).

Άσκηση 2 – Εναλλακτικές Υλοποιήσεις

```
class Ex1a {
  public static void main(String[] argv) {
      Double d1 = new Double(12.0);
      Double d2 = new Double(24.0);
      Double d3 = d1 + d2;
      System.out.println(d1 + "+" + d2 + "=" + d3);
      }
  }
```

```
class Ex1b {
   static Double d1, d2, d3;
   public static void main(String[] argv) {
       d1 = new Double(12.0);
       d2 = new Double(24.0);
       d3 = d1 + d2;
       System.out.println(d1 + "+" + d2 + "=" + d3);
       }
   }
```

```
class Ex1c {
   Double d1, d2, d3;
   public static void main(String[] argv) {
       Ex1c myObject = new Ex1c();
       myObject.d1 = new Double(12.0);
       myObject.d2= new Double(24.0);;
       myObject.d3 = myObject.d1 + myObject.d2;
       System.out.println(myObject.d1 + "+" +
                   myObject.d2 + "=" + myObject.d3);
       }
   }
```

```
class Ex1c2 {
   public static void main(String[] argv) {
       MyClass myObject = new MyClass();
       myObject.d1 = new Double(12.0);
       myObject.d2= new Double(24.0);
       myObject.d3 = myObject.d1 + myObject.d2;
       System.out.println(myObject.d1 + "+" +
                   myObject.d2 + "=" + myObject.d3);
       }
   }

class MyClass {
   Double d1, d2, d3;
   }
```


Δραστηριότητα 5.2 Object state

Δραστηριότητα 5.2 Κατάσταση Αντικειμένου (Object State)

Αναπτύξτε ένα πρόγραμμα `ObjectState` σύμφωνα με το οποίο το σύστημα θα δημιουργεί ένα αντικείμενο obj1 που θα έχει 2 καταστάσεις: ON και OFF. Η αρχική κατάσταση του obj1 τη στιγμή της δημιουργίας του είναι OFF.

Στη συνέχεια, το σύστημα θα δημιουργεί ένα άλλο αντικείμενο obj2 και θα του στέλνει το μήνυμα press. Η συμπεριφορά του obj2 στο μήνυμα press περιγράφεται στη συνέχεια.

α) Ζητάει από το obj1 να τυπώσει την κατάστασή του.

β) Ζητάει από το obj1 να αλλάξει την κατάστασή του σε ON.

γ) Ζητάει από το obj1 να τυπώσει την κατάστασή του.

δ) Ζητάει από το obj1 να αλλάξει την κατάστασή του σε OFF.

ε) Ενημερώνει τον χρήστη πως η συμπεριφορά στο μήνυμα press ολοκληρώθηκε επιτυχώς.

Τέλος, το σύστημα ενημερώνει τον χρήστη πως η εκτέλεση του προγράμματος `ObjectState` ολοκληρώθηκε.

Δώστε ένα πρόχειρο διάγραμμα κλάσεων για την εφαρμογή σας.

Object state – Identify Objects

Δραστηριότητα 5.2 Κατάσταση Αντικειμένου (Object State)

Αναπτύξτε ένα πρόγραμμα `ObjectState` σύμφωνα με το οποίο το σύστημα θα δημιουργεί ένα αντικείμενο obj1 που θα έχει 2 καταστάσεις: ON και OFF. Η αρχική κατάσταση του obj1 τη στιγμή της δημιουργίας του είναι OFF.

Στη συνέχεια, το σύστημα θα δημιουργεί ένα άλλο αντικείμενο obj2 και θα του στέλνει το μήνυμα press. Η συμπεριφορά του obj2 στο μήνυμα press περιγράφεται στη συνέχεια.

α) Ζητάει από το obj1 να τυπώσει την κατάστασή του.

β) Ζητάει από το obj1 να αλλάξει την κατάστασή του σε ON.

γ) Ζητάει από το obj1 να τυπώσει την κατάστασή του.

δ) Ζητάει από το obj1 να αλλάξει την κατάστασή του σε OFF.

ε) Ενημερώνει τον χρήστη πως η συμπεριφορά στο μήνυμα press ολοκληρώθηκε επιτυχώς.

© 2023 Κλεάνθης Θραμπουλίδης