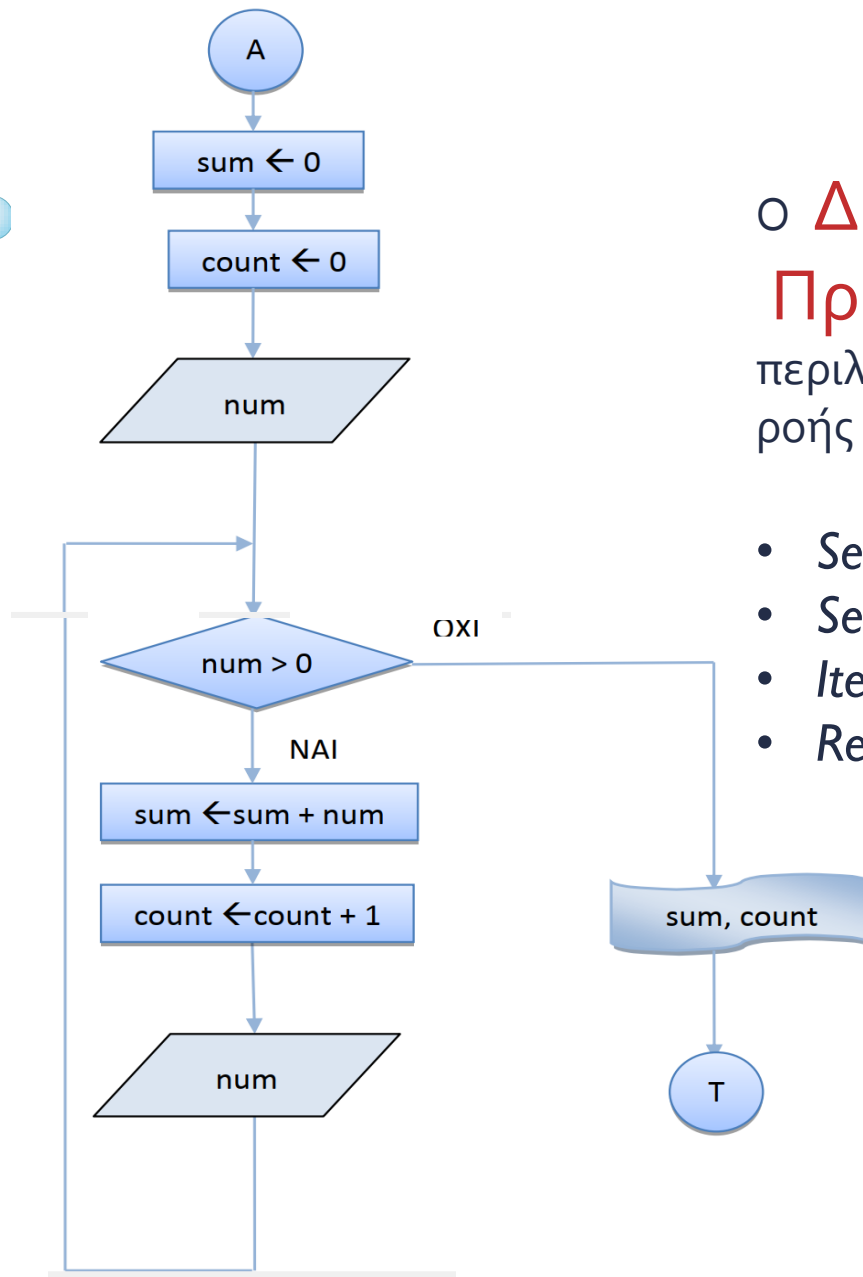




Αντικειμενοστρεφής Προγραμματισμός Εισαγωγή

Ασδρέ Κατερίνα
asdre@ihu.gr

- Διαδικαστικός Προγραμματισμός
- Αντικειμενοστρεφής Προγραμματισμός



ο Διαδικαστικός Προγραμματισμός

περιλαμβάνει άλλους τρόπους ελέγχου της ροής του προγράμματος:

- *Sequence* – ακολουθία εντολών
- *Selection* – επιλογή, if-then-else
- *Iteration* – εντολές επανάληψης
- *Recursion* – αναδρομή

Ο κώδικας οργανώνεται σε
διαδικασίες (procedures).

Ο **Αντικειμενοστρεφής Προγραμματισμός** βασίζεται σε αντικείμενα που αλληλεπιδρούν μεταξύ τους. Θυμίζει την οργάνωση και λειτουργία του πραγματικού κόσμου.

Ο πραγματικός κόσμος αποτελείται από αντικείμενα, υλικά ή μη, που η συμπεριφορά τους εξαρτάται από το περιβάλλον.

Διαδικαστικός VS Αντικειμενοστρεφής

Διαδικασίες και
συναρτήσεις που
επεξεργάζονται τα
δεδομένα είναι τα
δομικά στοιχεία των
προγραμμάτων

σκεφτόμαστε με κέντρο το
πρόβλημα,
χρησιμοποιούμε
αντικείμενα για
αναπαράσταση οντοτήτων
του προβλήματος, έχουμε
υψηλότερο επίπεδο
αφαίρεσης

Διαδικαστικός VS Αντικειμενοστρεφής

Ο προγραμματισμός στρέφεται προς τις έννοιες/αντικείμενα και όχι πια στις διαδικασίες.

Στα αντικείμενα ανατίθενται χαρακτηριστικά (*ιδιότητες*) που σχετίζονται με αυτά, τα οποία επεξεργάζεται με ειδικές συναρτήσεις (*μεθόδους*).

Αντικείμενο-Ιδιότητες-Μέθοδοι

- Αυτοκίνητο, άνθρωπος, ζώο, ζάρι, ...
- Πώς θα περιγράψαμε ένα αυτοκίνητο?
- Τι μπορούμε να κάνουμε με ένα αυτοκίνητο?
- Και πώς «δημιουργούμε» ένα αυτοκίνητο?



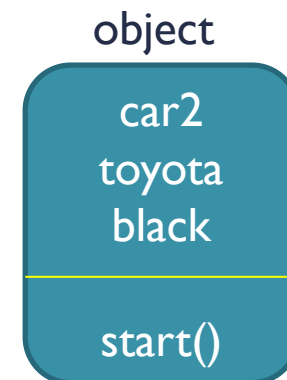
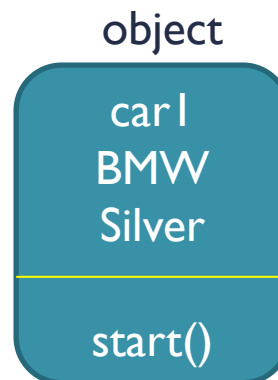
Δομητής (Constructor) – Κλάσεις Αντικειμένων

- Ειδική μέθοδος δημιουργίας αντικειμένων: **constructor**
- Αρχικοποίηση (μάρκα, χλμ=0, ...)
- Αν θέλω να κατασκευάσω και φορτηγό? Ή ποδήλατο?
- Θα περιγράψω το καθένα ξεχωριστά ... έχουν όμως κοινά χαρακτηριστικά! Γι' αυτό δημιουργώ **πρότυπα** που συγκεντρώνουν τα κοινά χαρακτηριστικά και τις κοινές μεθόδους.
- Τα πρότυπα αυτά (σχέδια) ονομάζονται **classes**.



Δομητής (Constructor) – Κλάσεις Αντικειμένων

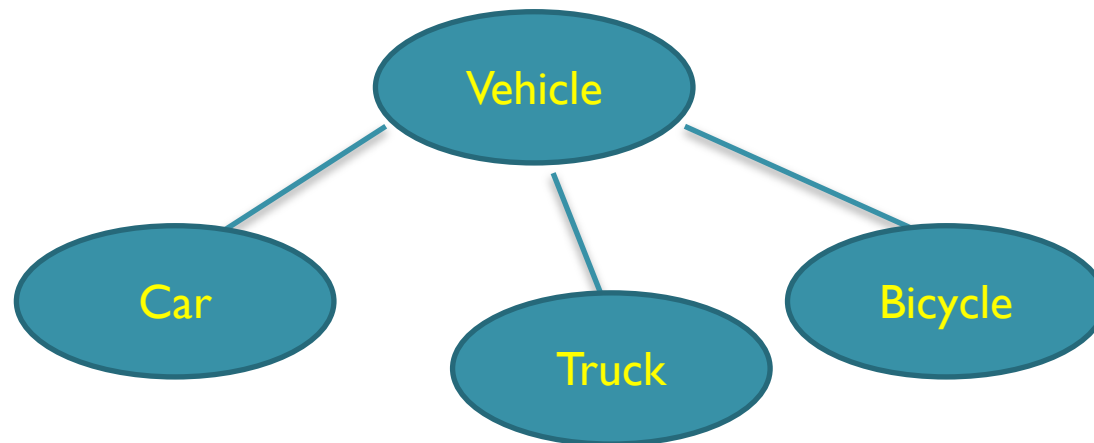
- Από την κλάση γνωρίζω για το αντικείμενο πριν τη δημιουργία του.
- Ουσιαστικά η κλάση ορίζει έναν νέο τύπο δεδομένων.
- Στις κλάσεις εμπεριέχονται και οι δομητές.
- Έχοντας τον τρόπο για να το φτιάξω, δεν σημαίνει ότι το έχω κι όλες!!
- Ένα αντικείμενο είναι ένα στιγμιότυπο της κλάσης.



Κλάσεις Αντικειμένων - Κληρονομικότητα

- Άρα για το αυτοκίνητο, το φορτηγό και το ποδήλατο θα μπορούσα να έχω μια κλάση **vehicle** (κλάση γονέας) η οποία θα περιέχει τα κοινά χαρακτηριστικά και τις μεθόδους των τριών (υποκλάσεις).
- Οι υποκλάσεις κληρονομούν ιδιότητες και μεθόδους.
- Αφού κληρονομούνται δεν τα ξαναορίζω. Μόνο ό,τι δεν υπάρχει στην υπερκλάση.

Κλάσεις Αντικειμένων - Κληρονομικότητα



Αρχές Αντικειμενοστρεφούς Προγραμματισμού

- ☐ Ενθυλάκωση
- ☐ Αφαίρεση
- ☐ Κληρονομικότητα
- ☐ Πολυμορφισμός

Αρχές Αντικειμενοστρεφούς Προγραμματισμού Ενθυλάκωση

Οι λεπτομέρειες της υλοποίησης κρυμμένες από
το χρήστη του αντικειμένου.

Αρχές Αντικειμενοστρεφούς Προγραμματισμού Αφαίρεση

Αποτέλεσμα της ενθυλάκωσης.

Δηλώνουμε τις λειτουργίες χωρίς να μας

ενδιαφέρει ο τρόπος υλοποίησης. Απλοποίηση...

Αρχές Αντικειμενοστρεφούς Προγραμματισμού Πολυμορφισμός

Μία μέθοδος υλοποιείται διαφορετικά ανάλογα
με το αντικείμενο που την χρησιμοποιεί.

Υπερφόρτωση VS Υπέρβαση



Java

- δημιουργήθηκε το 1995 από τη Sun Microsystems, την οποία το 2010 εξαγόρασε η Oracle
- θεωρείται μία από τις πιο δημοφιλείς γλώσσες και από το 2006 είναι γλώσσα ανοιχτού κώδικα όσον αφορά στο μεταγλωττιστή (javac) και το πακέτο ανάπτυξης (JDK, Java Development Kit)



Java

- **Write Once, Run Anywhere:** μπορεί να εκτελείται σε όλες τις πλατφόρμες, τρέχουν το ίδιο σε Windows, Linux, Unix, Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση ή να αλλάξει κάτι στον πηγαίο κώδικα.
- όταν ένα πρόγραμμα σε Java μεταγλωττίζεται, αυτό που παράγεται δεν είναι κώδικας μηχανής για έναν πραγματικό επεξεργαστή, αλλά ένα είδος κώδικα γνωστό ως **java bytecode**, που προορίζεται για μια εικονική μηχανή, **Java Virtual Machine**. Ο κώδικας αυτός εκτελείται από το περιβάλλον εκτέλεσης της Java (Java Runtime Environment, JRE), που είναι διαφορετικό για κάθε λειτουργικό σύστημα και είναι **διερμηνευτής** (interpreter).



Java

Μεταγλώττιση και εκτέλεση προγραμμάτων

- όταν ένα πρόγραμμα σε Java μεταγλωττίζεται, αυτό που παράγεται δεν είναι κώδικας μηχανής για έναν πραγματικό επεξεργαστή, αλλά ένα είδος κώδικα γνωστό ως **java bytecode**, που προορίζεται για μια εικονική μηχανή, **Java Virtual Machine**. Ο κώδικας αυτός εκτελείται από το περιβάλλον εκτέλεσης της Java (Java Runtime Environment, JRE), που είναι διαφορετικό για κάθε λειτουργικό σύστημα και είναι διερμηνευτής (interpreter).





Java

Το πρώτο μου πρόγραμμα...

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



Java

Το πρώτο μου πρόγραμμα...περιέχει κλάση...

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

- Όνομα αρχείου: **HelloWorld.java** αφού μέσα στο πρόγραμμα μας έχουμε μια public κλάση με το όνομα HelloWorld.
- Η κλάση θα πρέπει να περιέχει μια μέθοδο main από την οποία θα ξεκινήσει η εκτέλεση του προγράμματος μας.
- String: κλάση για χειρισμό αλφαριθμητικών.



Java

Το πρώτο μου πρόγραμμα...περιέχει κλάση...

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

- public, static
- System.out: αντικείμενο
- println(): μέθοδος που τυπώνει το String αντικείμενο που δίνεται ως όρισμα και αλλάζει γραμμή.
- Ομοιότητες με την C?



Java

Το δεύτερό μου πρόγραμμα...

Να γραφτεί πρόγραμμα που αθροίζει δύο ακραίους

```
public class Sum2Integers
{
    public static void main(String[] args)
    {
        int num1 = 3; //or int num1=3, num2=10;
        int num2 = 10;
        int result = num1 + num2;
        System.out.println("Result = " + result);
    }
}
```



Java

Το δεύτερό μου πρόγραμμα...

Να γραφτεί πρόγραμμα που αθροίζει δύο ακραίους

```
public class Sum2Integers
{
    public static void main(String[] args)
    {
        int num1 = 3; //or int num1=3, num2=10;
        int num2 = 10;
        int result = num1 + num2;
        System.out.println("Result = " + result);
    }
}
```

- τελεστής «+»: ο βασικός τύπος (int) μετατρέπεται σε String και γίνεται η συνένωση με το String **Result =** "



Java

Βασικοί τύποι (πρωτογενείς, δεν είναι αντικείμενα κάποιας κλάσης)

Τύπος	Μέγεθος	Εύρος
<code>byte</code>	1 byte	-128 έως 127
<code>short</code>	2 bytes	-32768 έως 32767
<code>int</code>	4 bytes	-2^{31} έως $2^{31}-1$
<code>long</code>	8 bytes	-2^{63} έως $2^{63}-1$
<code>float</code>	4 bytes	-3.40292347E+38 έως 3.40292347E+38
<code>double</code>	8 bytes	-1.79769313486231570E+308 έως 1.79769313486231570E+308
<code>boolean</code>	1 bytes	true / false
<code>char</code>	2 bytes	οποιοδήποτε γράμμα ή ψηφίο ή άλλο σύμβολο του κώδικα <i>unicode</i>



Java

Τελεστές

Αριθμητικοί Τελεστές	Σχεσιακοί Τελεστές	Λογικοί Τελεστές
+	==	!
-	<	&&
*	>	
/	<=	^
%	>=	
	!=	



Java

Το τρίτο μου πρόγραμμα...

```
public class Grade{  
    static double FGrade(double lab, double theory){  
        double result;  
        result=0.3 * lab + 0.7 * theory;  
        return result;  
    }  
    public static void main (String argv[]){  
        double lab=6.5, theory=5.4, apot;  
        apot=FGrade(lab,theory);  
        System.out.println("Grade = "+apot);  
    }  
}
```



Java

Το τελευταίο για σήμερα πρόγραμμα...

```
public class Student{
    private int am;
    private String onoma;
    private int apousies;
    private double vathmos;

    public Student(){}
    public Student(int am,String onoma){
        this(am, onoma, 0,0.0);
    }
    public Student(int am,String onoma,int apousies,
        double vathmos){

        this.am=am;
        this.onoma=onoma;
        this.apousies=apousies;
        this.vathmos=vathmos;
    }
}
```



Java

Το τελευταίο για σήμερα πρόγραμμα...

```
public void setVathmos(double Vathmos) {  
    vathmos=Vathmos;  
}
```

```
public void setApousies(int apousies) {  
    this.apousies=apousies;  
}
```

```
public void setAm(int am) {  
    this.am=am;  
}
```

```
public void setOnoma(String onoma) {  
    this.onoma=onoma;  
}
```



Java

Το τελευταίο για σήμερα πρόγραμμα...

```
public int getAM(){  
    return am;  
}  
  
public String getOnoma(){  
    return onoma;  
}  
  
public double getVathmos(){  
    return vathmos;  
}  
  
public int getApousies(){  
    return apousies;  
}
```



Java

Το τελευταίο για σήμερα πρόγραμμα...

```
public String toString(){  
    return ("AM: "+am+"\nOnoma: "+onoma+"\nVathmos: "+  
            vathmos+"\nApousies: "+apousies);  
}
```

```
public static void main(String[] args) {  
    Student S1=new Student(181001, "Nikos");  
    System.out.println(S1);  
    S1.onoma="Katerina";  
    System.out.println(S1.toString());  
}
```

```
}
```



Java

Το τελευταίο για σήμερα πρόγραμμα...

```
public String toString(){  
    return ("AM: "+am+"\nOnoma: "+onoma+"\nVathmos: "+  
        vathmos+"\nApousies: "+apousies);  
}
```

```
public static void main(String[] args) {  
    Student S1=new Student(181001, "Nikos");  
    //or Student S1;  
    //and S1=new Student(181001, "Nikos");  
    System.out.println(S1);  
    S1.onoma="Katerina";  
    //S1.setOnoma("Katerina");  
    System.out.println(S1.toString());  
}
```

```
}
```