Υπερφόρτωση Μεθόδων Πολυμορφισμός Κληρονομικότητα

Παραδείγματα

Ε.Ι.Βογιατζάκη

Constructors Υπερφόρτωση
ConstructorExample

```
public class Student
{
    public static int id = 0;
    public int age;
    public String firstName;
    public String lastName;
    public String address;


    public Student()
    {       id++;       }

    public Student(String fname, String lname)  {
        this();
        this.firstName = fname;
        this.lastName = lname;     }

    public Student(String fname, String lname, int age) {
        this(fname, lname);
        this.age = age;     }

    public Student(String fname, String lname, String address) {
        this(fname, lname);
        this.address = address;     }

}
```

```
public class Student
{
    public static int id = 0;
    public int age;
    public String firstName;
    public String lastName;
    public String address;

    public Student()
    {       id++;       }

    public Student(String fname, String lname)  {
        this();
        this.firstName = fname;
        this.lastName = lname;     }

    public Student(String fname, String lname, int age) {
            this(fname, lname);
            this.age = age;     }

    public Student(String fname, String lname, String address) {
            this(fname, lname);
            this.address = address;     }

}
```

```
public class StudentExample
{
  public static void main(String[] args)
  {
     Student s1 = new Student();
     Student s2 = new Student("Kostas", "Georgiou");
     Student s3 = new Student("Maria", "Spinou", 20);
     Student s4 = new Student("Katerina", "Kiriakidou", "Panepistimiou 2, 26100, Patra");



     System.out.println("1st Student: " + s1.firstName + " " + s1.lastName);
     System.out.println("2nd Student: " + s2.firstName + " " + s2.lastName);
     System.out.println("3rd Student: " + s3.firstName + " " + s3.lastName + " " + s3.age);
     System.out.println("4th Student: " + s4.firstName + " " + s4.lastName + " " +
s4.address);

              System.out.println("\nNumber of students: " + Student.id); //access static
variable id
 }
}
```

```
1st Student: null null
2nd Student: Kostas Georgiou
3rd Student: Maria Spinou 20
4th Student: Katerina Kiriakidou Panepistimiou 2, 26100, Patra

Number of students: 4
```

```
public class Student
{
    public static int id = 1;
    public int age;
    public String firstName;
    public String lastName;
    public String address;

    public Student()
    {       id++;       }

    public Student(String fname, String lname)  {
        this();
        this.firstName = fname;
        this.lastName = lname;     }
```

Τι θα συμβεί αν το static int id =1 αντι για 0

```
 public Student(String fname, String lname, int
age) {
        this(fname, lname);
        this.age = age;     }

    public Student(String fname, String lname,
String address) {
        this(fname, lname);
        this.address = address;     }

}
```

```
public class StudentExample
{
  public static void main(String[] args)
  {
     Student s1 = new Student();
     Student s2 = new Student("Kostas", "Georgiou");
     Student s3 = new Student("Maria", "Spinou", 20);
     Student s4 = new Student("Katerina", "Kiriakidou", "Panepistimiou 2, 26100, Patra");


     System.out.println("1st Student: " + s1.firstName + " " + s1.lastName);
     System.out.println("2nd Student: " + s2.firstName + " " + s2.lastName);
     System.out.println("3rd Student: " + s3.firstName + " " + s3.lastName + " " + s3.age);
     System.out.println("4th Student: " + s4.firstName + " " + s4.lastName + " " +
s4.address);

              System.out.println("\nNumber of students: " + Student.id); //access static
variable id
 }
}
```

```
1st Student: null null
2nd Student: Kostas Georgiou
3rd Student: Maria Spinou 20
4th Student: Katerina Kiriakidou Panepistimiou 2, 26100, Patra

Number of students: 5
```

```
public class StudentExample
{
  public static void main(String[] args)
  {
     Student s1 = new Student();
     Student s2 = new Student("Kostas", "Georgiou");
     Student s3 = new Student("Maria", "Spinou", 20);
     Student s4 = new Student("Katerina", "Kiriakidou", "Panepistimiou 2, 26100, Patra");


     //System.out.println("1st Student: " + s1.firstName + " " + s1.lastName);
     System.out.println("2nd Student: " + s2.firstName + " " + s2.lastName);
     System.out.println("3rd Student: " + s3.firstName + " " + s3.lastName + " " + s3.age);
     System.out.println("4th Student: " + s4.firstName + " " + s4.lastName + " " +
s4.address);

              System.out.println("\nNumber of students: " + Student.id); //access static
variable id
 }
}
```

Τι θα συμβεί αν το static int id =1
αντί για 0
και δεν τρέξει το
system.out.prinln για το s1?

```
2nd Student: Kostas Georgiou
3rd Student: Maria Spinou 20
4th Student: Katerina Kiriakidou Panepistimiou 2, 26100, Patra

Number of students: 5
```

```
public class StudentExample
{
  public static void main(String[] args)
  {
     //Student s1 = new Student();
     Student s2 = new Student("Kostas", "Georgiou");
     Student s3 = new Student("Maria", "Spinou", 20);
     Student s4 = new Student("Katerina", "Kiriakidou", "Panepistimiou 2, 26100, Patra");



     //System.out.println("1st Student: " + s1.firstName + " " + s1.lastName);
     System.out.println("2nd Student: " + s2.firstName + " " + s2.lastName);
     System.out.println("3rd Student: " + s3.firstName + " " + s3.lastName + " " + s3.age);
     System.out.println("4th Student: " + s4.firstName + " " + s4.lastName + " " +
s4.address);

              System.out.println("\nNumber of students: " + Student.id); //access static
variable id
 }
}
```

Τι θα συμβεί αν το static int id =1
αντί για 0
και δεν δημιουργηθεί το s1?

```
2nd Student: Kostas Georgiou
3rd Student: Maria Spinou 20
4th Student: Katerina Kiriakidou Panepistimiou 2, 26100, Patra

Number of students: 4
```

exampleGetSet

```
public class Student
{
   public static int id = 0;
   private final int entryYear;
   private static final String DOMAIN = "@ceid.upatras.gr";
   private String firstName;
   private String lastName;
   private String email;

   public Student(int year)
   {
      id++;
      entryYear = year;

   }

   public Student(String fname, String lname, int year)
   {
      this(year);
      firstName = fname;
      lastName = lname;
   }
```

```
// Set value for the private variables firstName and lastName
   public void setName(String fname, String lname)
   {
      setFirstName(fname);
      setLastName (lname); }

   private void setFirstName (String fname) {
      this.firstName = fname;}

   private void setLastName (String lname) {
      this.lastName = lname;
   }

   // Get the value of the private variables firstName and
lastName
   public String getName()
   {
      String fullName = this.firstName + " " + this.lastName;
      return fullName;
   }

   public String getEmail(){
      email = this.lastName + "_" + entryYear + DOMAIN;
      return email;}
}
```

Προσπάθεια εκτύπωσης του private String firstName χωρίς χρήση μεθόδου getName

```
public class SetterGetterExample
{

   public static void main(String[] args)
   {
       Student s1 = new Student("Kostas", "Georgiou", 2010);
       Student s2 = new Student("Maria", "Spinou", 2008);
       Student s3 = new Student("Katerina", "Kiriakidou", 2005);

       //firstName and lastName are private fields
      System.out.println("1st Student: " + s1.firstName + " " + s1.lastName);
       System.out.println("1st Student: " + s
       System.out.println("2nd Student: " + s
       System.out.println("Email: " + s2.getEmail());
       //setFirstName is a private method
      //s3.setFirstName ("Eleni");
      s3.setName ("Eleni", "Kiriakidou");
       System.out.println("3rd Student: " + s3.getName());
       System.out.println("\nNumber of students: " + s1.id); //access static var
   }
}
```

firstName has private access in Student

Ορισμός τιμής του private String firstName με την private μέθοδο setFirstName

```
public class SetterGetterExample
{

   public static void main(String[] args)
   {
       Student s1 = new Student("Kostas", "Georgiou", 2010);
       Student s2 = new Student("Maria", "Spinou", 2008);
       Student s3 = new Student("Katerina", "Kiriakidou", 2005);

       //firstName and lastName are private fields
      //System.out.println("1st Student: " + s1.firstName + " " + s1.lastName);
       System.out.println("1st Student: " + s1.getName());
       System.out.println("2nd Student: " + s2.getName());
       System.out.println("Email: " + s2.getEmail());
       //setFirstName is a private method
       s3.setFirstName ("Eleni");
       s3
       Sy
       System.out.println("\nNumber of students: " + s1.id); //access static variable id
   }
}
```

setFirstName(java.lang.String) has private access in Student

```
public class SetterGetterExample
{

  public static void main(String[] args)
  {
     Student s1 = new Student("Kostas", "Georgiou", 2010);
     Student s2 = new Student("Maria", "Spinou", 2008);
     Student s3 = new Student("Katerina", "Kiriakidou", 2005);

     //firstName and lastName are private fields
    //System.out.println("1st Student: " + s1.firstName + " " + s1.lastName);
     System.out.println("1st Student: " + s1.getName());
     System.out.println("2nd Student: " + s2.getName());
     System.out.println("Email: " + s2.getEmail());
     //setFirstName is a private method
    //s3.setFirstName ("Eleni");
    s3.setName ("Eleni", "Kiriakidou");
     System.out.println("3rd Student: " + s3.getName());
     System.out.println("\nNumber of students: " + s1.id); //access static variable id
  }
}
```

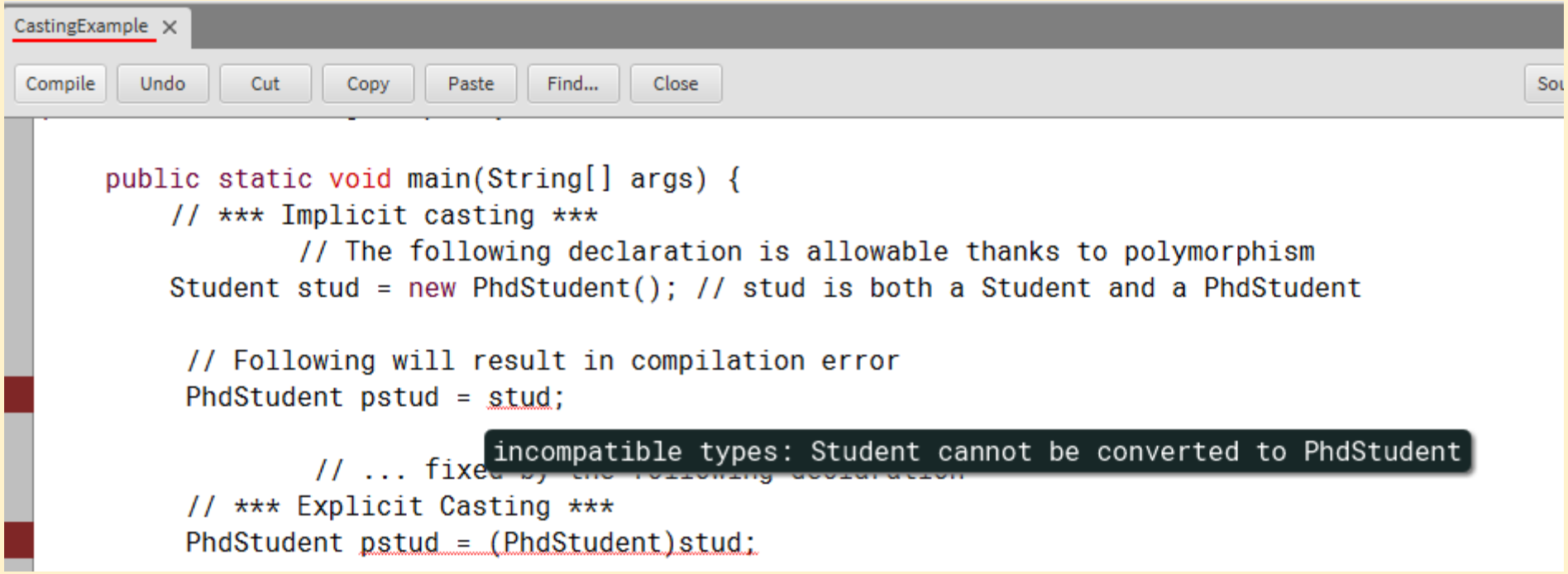
Εκτύπωση του private String firstName με τη μέθοδο getName
Ορισμός τιμής του με την public μέθοδο setName

```
Number of students: 3
1st Student: Kostas Georgiou
2nd Student: Maria Spinou
Email: Spinou_2008@ceid.upatras.gr
3rd Student: Eleni Kiriakidou
```

Πολυμορφισμός, CastingExample

```
class Student {
}

class UnderGraduateStudent extends Student {

   void underMethod() {
      System.out.println("This is a undergraduateMethod defined in Class UnderGraduateStudent");
   }
}

class MasterStudent extends Student {
   void phdMethod() {
      System.out.println("This is a phdMethod defined in Class MasterStudent");
   }

}

class PhdStudent extends Student {

   void phdMethod() {
      System.out.println("This is a phdMethod defined in Class PhdStudent");
   }
}
```

Πολυμορφισμός, CastingExample



Πολυμορφισμός, CastingExample

CastingExample - CastingExamplePrim

Class Edit Tools Options

CastingExample

Compile Undo Cut Copy Paste Find... Close Source Code

```
public class CastingExample {

    public static void main(String[] args) {
        // *** Implicit casting ***
                // The following declaration is allowable thanks to polymorphism
        Student stud = new PhdStudent(); // stud is both a Student and a PhdStudent

         // Following will result in compilation error
         //PhdStudent pstud = stud;

                 // ... fixed by the following declaration
         // *** Explicit Casting ***
         //PhdStudent pstud = (PhdStudent)stud;

         // Following will result in run-time error
         MasterStudent mstud = (MasterStudent)stud; // Cannot convert as they are sibblings

         // Following will result in compilation error

         // stud.phdMethod();  //As the method phdMethod is undefined for the Student Type
```

java.lang.ClassCastException:
class PhdStudent cannot be cast to class MasterStudent (PhdStudent and MasterStudent are in unnamed module of loader java.net.URLClassLoader @6630342f)

saved

Σε επίπεδο εκτέλεσης εμφανίζεται error

Πολυμορφισμός, CastingExample

```
    public static void main(String[] args) {
        // *** Implicit casting ***
                // The following declaration is allowable thanks to polymorphism
        Student stud = new PhdStudent(); // stud is both a Student and a PhdStudent

         // Following will result in compilation error
         //PhdStudent pstud = stud;

                 // ... fixed by the following declaration
         // *** Explicit Casting ***
         PhdStudent pstud = (PhdStudent)stud;

         // Following will result in run-time error
         // MasterStudent mstud = (MasterStudent)stud; // Cannot convert as they are si

         // Following will result in compilation error

          stud.phdMethod();  //As the method phdMethod is undefined for the Student Typ
                cannot find symbol -   method phdMethod()
         // ((PhdStudent)stud).phdMethod();

    }
}
```

Η μέθοδος phdMethod() δεν μπορεί να κληθεί από το stud.

Πολυμορφισμός, CastingExample

```
    public static void main(String[] args) {
        // *** Implicit casting ***
                // The following declaration is allowable thanks to polymorphism
        Student stud = new PhdStudent(); // stud is both a Student and a PhdStudent

         // Following will result in compilation error
         //PhdStudent pstud = stud;

                 // ... fixed by the following declaration
         // *** Explicit Casting ***
         PhdStudent pstud = (PhdStudent)stud;

         // Following will result in run-time error
         // MasterStudent mstud = (MasterStudent)stud; // Cannot convert as they are si

         // Following will result in compilation error

          stud.phdMethod();  //As the method phdMethod is undefined for the Student Typ
                cannot find symbol -   method phdMethod()
         // ((PhdStudent)stud).phdMethod();

    }
}
```

Η μέθοδος phdMethod()
δεν μπορεί
να κληθεί από το stud.

Πολυμορφισμός, CastingExample

CastingExample

Compile | Undo | Cut | Copy | Paste | Find... | Close | Source Cod

```
public class CastingExample {

    public static void main(String[] args) {
        // *** Implicit casting ***
                // The following declaration is allowable thanks to polymorphism
        Student stud = new PhdStudent(); // stud is both a Student and a PhdStudent

         // Following will result in compilation error
         //PhdStudent pstud = stud;

                 // ... fixed by the following declaration
         // *** Explicit Casting ***
         PhdStudent pstud = (PhdStudent)stud;

         // Following will result in run-time error
         // MasterStudent mstud = (MasterStudent)stud; // Cannot convert as they are sibblings

         // Following will result in compilation error

         // stud.phdMethod();  //As the method phdMethod is undefined for the Student Type

                 // ... fixed by doing casting
         ((PhdStudent)stud).phdMethod();

    }
```

Διόρθωση με casting

Class compiled - no syntax errors

Πολυμορφισμός, CastingExample

```
public class CastingExample {

   public static void main(String[] args) {
      // *** Implicit casting ***
                        // The following declaration is allowable thanks to polymorphism
      Student stud = new PhdStudent(); // stud is both a Student and a PhdStudent

       // Following will result in compilation error
       //PhdStudent pstud = stud;

                         // ... fixed by the following declaration
       // *** Explicit Casting ***
       PhdStudent pstud = (PhdStudent)stud;

       // Following will result in run-time error
       // MasterStudent mstud = (MasterStudent)stud; // Cannot convert as they are sibblings

       // Following will result in compilation error
       // stud.phdMethod();  //As the method phdMethod is undefined for the Student Type

                         // ... fixed by doing casting
       // ((PhdStudent)stud).phdMethod();

   }
}
```

This is a phdMethod defined in Class PhdStudent

OverloadingExample
Μέθοδοι Υπερφόρτωση

```
class Student {

   void studMethod() {
      System.out.println ("The no arguments studMethod of class Student has been called");
   }

   void studMethod(String name, String surname){
      System.out.println ("The two arguments studMethod of class Student has been called");
      String fullName = name + " " + surname;
      System.out.println("Full name: " + fullName);
   }

}

/* MasterStudent is subclass of Student */
class MasterStudent extends Student {

   String studMethod(String name, String surname, int id)  {
      super.studMethod(name, surname);
      System.out.println("This is a master student.");
      String  sid = "master_" + id;
      return sid;
   }
}
```

```
public class OverloadingExample {

    public static void main(String[] args) {

        Student stud1 = new Student();
        Student stud2 = new Student();
        MasterStudent mstud = new MasterStudent();

        System.out.println("\nInfo about 1st student:");
        stud1.studMethod();

        System.out.println("\nInfo about 2nd student:");
        stud2.studMethod("Maria", "Georgiou");

        System.out.println("\nInfo about 3rd student:");
        String mID = mstud.studMethod("Nikos", "Oikonomou", 22);
        System.out.println("Student ID: " + mID);
    }
}
```

```
Info about 1st student:
The no arguments studMethod of class Student has been called

Info about 2nd student:
The two arguments studMethod of class Student has been called
Full name: Maria Georgiou

Info about 3rd student:
The two arguments studMethod of class Student has been called
Full name: Nikos Oikonomou
This is a master student.
Student ID: master_22
```

OverridingExample

```
class Student {
   String studMethod(int id) {
      System.out.println ("\nThe studMethod of class Student has been
called.");
      String sid = "general_student_" + id;
      return sid;
   }

            final void printUniv(){
                        System.out.println("\nUniversity of Patras");
            }
}

/* UnderGraduateStudent is subclass of Student */
class UnderGraduateStudent extends Student {

   String studMethod(int id) {
      super.studMethod(id);
      System.out.println(" This is an under graduate student.");
      String sid = "bachelor_" + id;
      return sid;
   }
   //public void printUniv(){
            //            System.out.println("University of Patras");
            //}  // τι θα συμβεί αν βγεί το σχόλιο; compilation error
              //η printUniv είναιfinal

}
```

```
/* MasterStudent is subclass of Student */
class MasterStudent extends Student {
   String studMethod(int id) {
      super.studMethod(id);
      System.out.println(" This is a master student.");
      String sid = "master_" + id;
      return sid;
   }
}
```

OverridingExample

```
public class OverridingExample {

   public static void main(String[] args) {
      String studID;

      Student stud = new Student();
      //stud.printUniv();
      studID = stud.studMethod(1);
      System.out.println("Student ID: " + studID);

      UnderGraduateStudent stud1 = new UnderGraduateStudent();
      //stud1.printUniv();
      studID = stud1.studMethod(9999);
      System.out.println("Student ID: " + studID);

      MasterStudent stud2 = new MasterStudent();
      //stud2.printUniv();
      studID = stud2.studMethod(100);
      System.out.println("Student ID: " + studID);
   }
}
```

```
The studMethod of class Student has been called.
Student ID: general_student_1

The studMethod of class Student has been called.
 This is an under graduate student.
Student ID: bachelor_9999

The studMethod of class Student has been called.
 This is a master student.
Student ID: master_100
```

OverridingExample

```
public class OverridingExample {

   public static void main(String[] args) {
      String studID;

      Student stud = new Student();
      stud.printUniv();
      studID = stud.studMethod(1);
      System.out.println("Student ID: " + studID);

      UnderGraduateStudent stud1 = new UnderGraduateStudent();
      stud1.printUniv();
      studID = stud1.studMethod(9999);
      System.out.println("Student ID: " + studID);

      MasterStudent stud2 = new MasterStudent();
      stud2.printUniv();
      studID = stud2.studMethod(100);
      System.out.println("Student ID: " + studID);
   }
}
```

```
University of Patras

The studMethod of class Student has been called.
Student ID: general_student_1

University of Patras

The studMethod of class Student has been called.
 This is an under graduate student.
Student ID: bachelor_9999

University of Patras

The studMethod of class Student has been called.
 This is a master student.
Student ID: master_100
```

```
class Student {
   String studMethod(int id) {
      System.out.println ("\nThe studMethod of class Student
has been called.");
      String sid = "general_student_" + id;
      return sid;
   }

         final void printUniv(){
                  System.out.println("University of Patras");
         }
}
```

```
/* UnderGraduateStudent is subclass of Student */
class UnderGraduateStudent extends Student {

    String studMethod(int id) {
        super.studMethod(id);
        System.out.println(" This is an under graduate student.");
        String sid = "bachelor_" + id;
        return sid;
    }
    public void printUniv(){
                System.out.println("University of Patras");
        }  // τι θα συμβεί αν βγεί το σχόλιο; compilation error
           //η printUniv είναιfinal

}
```

```
printUniv() in UnderGraduateStudent cannot override
printUniv() in Student
  overridden method is final
```