Πολυμορφισμός Κληρονομικότητα

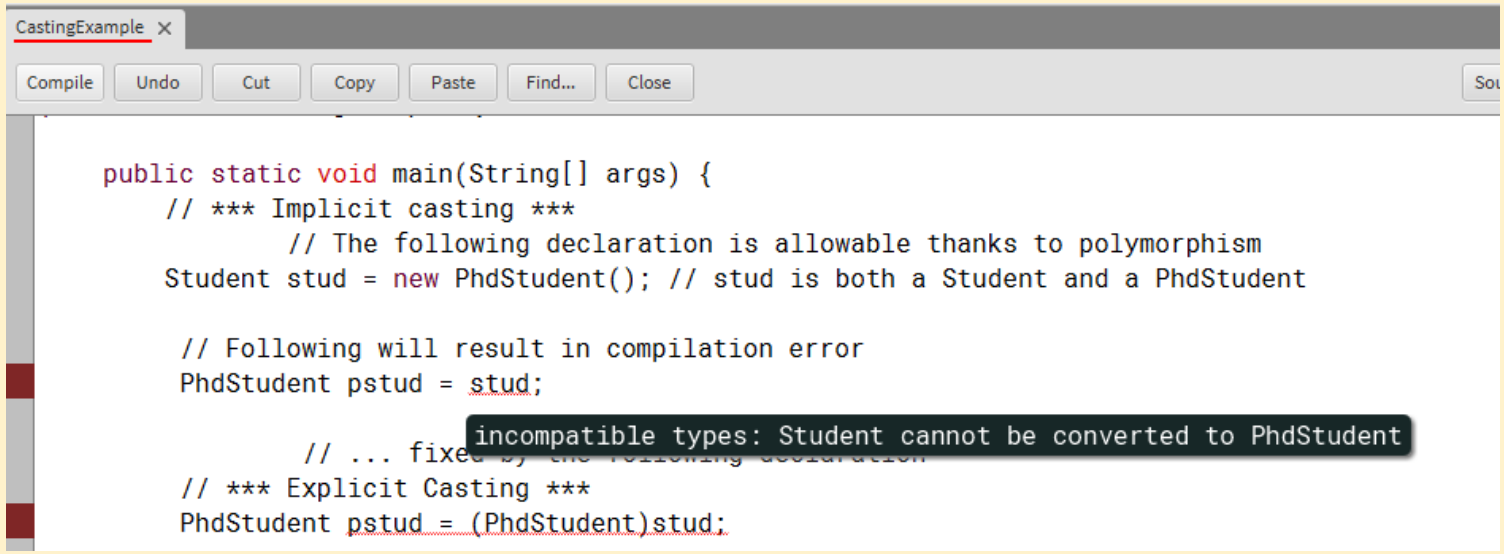
Παραδείγματα

Ε.Ι.Βογιατζάκη

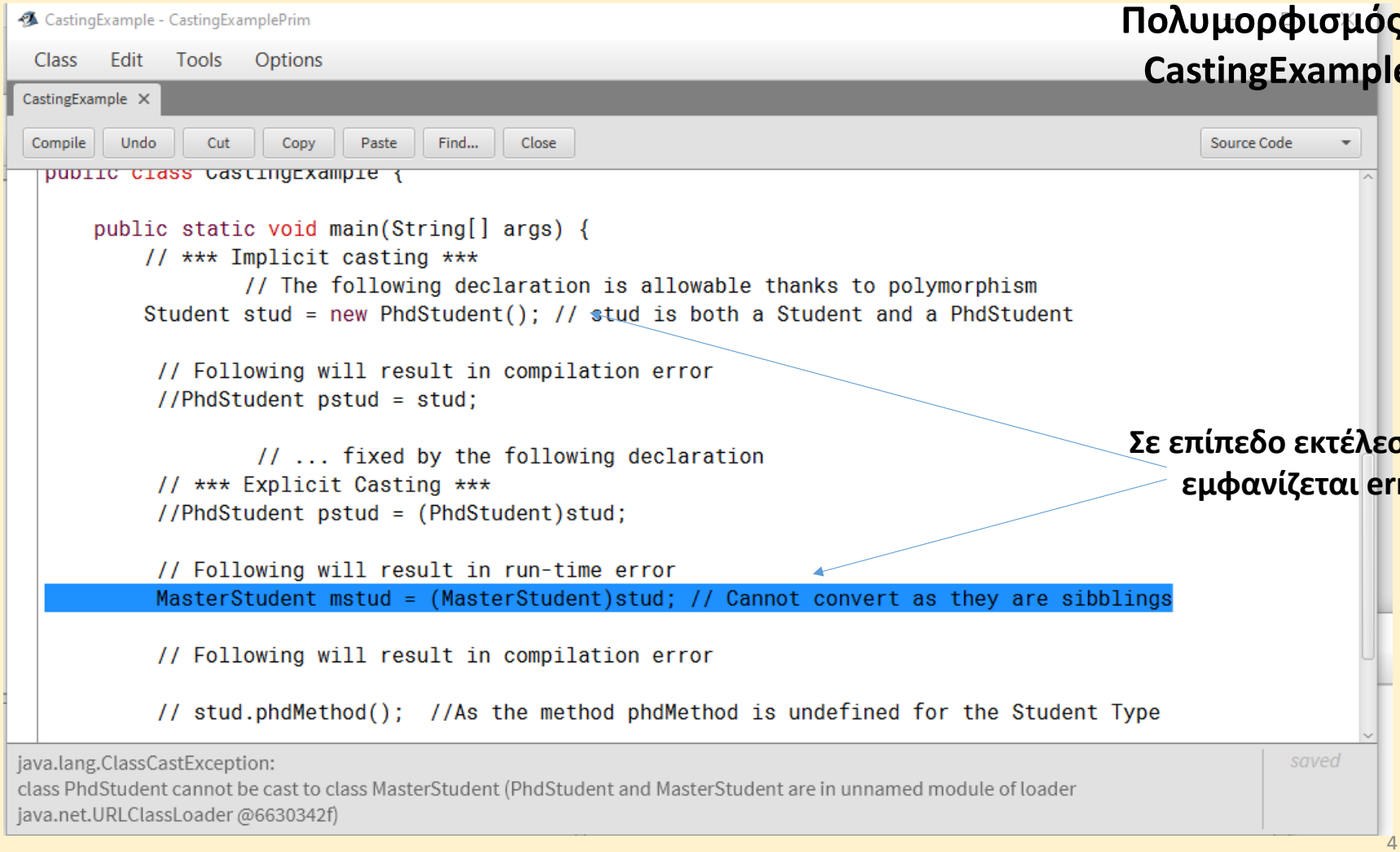
Πολυμορφισμός, CastingExample

```
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



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 ty
               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 ty
               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
```