



Fondamenti di Java

Fac.Scienze - Università di Trento

Upcast e Downcast
Ancora Polimorfismo
Abstract
UML



Sezione: Upcast - downcast

Fac.Scienze - Università di Trento

Upcast & downcast

Documentazione 2 - Marco Ronchetti

```

public class Test {
    public static void main(String a[]) {
        new Test();
    }

    Test() {
        A a;
        B b = new B();
        a=b;
        a.f1();
        ((C)a).f3();
    }
}

```

cast

OK: upcast implicito

NO: downcast illecito (runtime)
java.lang.ClassCastException

```

class A { void f1() { System.out.println("f1"); } }
class B extends A { void f2() { System.out.println("f2"); } }
class C extends B { void f3() { System.out.println("f3"); } }

```

Fac. Scienze - Università di Trento

6 Programmazione 2 - Marco Ronchetti

Type conversion - cast

Intro
JAVA

Si può applicare cast SOLO all'interno di una gerarchia di ereditarietà

È consigliabile usare l'operatore **instanceof** per verificare prima effettuare un downcast

```

if (staff[1] instanceof Manager) {
    Manager n = (Manager)staff[1];
    ...
}

```

Fac. Scienze - Università di Trento



La Pila in Java - 8a

```
public static void main(String args[]) {  
    int dim=10;  
    Pila s=new Pila();  
    //INSERIMENTO  
    for (int k=0;k<dim;k++){  
        Object o;  
        if (Math.random()<0.5)  
            o=new Integer(k);  
        else  
            o=new Float(k*Math.PI);  
        s.inserisci(o);  
    }  
}
```



La Pila in Java - 8b

```
// ESTRAZIONE  
for (int k=0;k<dim;k++) {  
    Object o = s.estrail();  
    if (o instanceof Integer) {  
        Integer i = (Integer) o;  
        int w = i.intValue();  
        System.out.println("an int:"+w);  
    } else if (o instanceof Float) {  
        Float i = (Float) o;  
        float w = i.floatValue();  
        System.out.println("a float:"+w);  
    } else  
        System.out.println("Unknown class!");  
}  
}
```



La Pila in Java - 8c

OUTPUT :

```
a float:28.274334
an int:8
an int:7
a float:18.849556
an int:5
an int:4
a float:9.424778
a float:6.2831855
a float:3.1415927
a float:0.0
```



Fondamenti di Java

Polimorfismo a tutto campo,
con Pile e Code...



Trasformare la Pila in Coda

```
package strutture;
public class Coda extends Pila{
    Object estrai() {
        assert(marker>0):"Estrazione da Coda vuota";
        Object retval=contenuto[0];
        for (int k=1; k<marker; k++ )
            contenuto[k-1]=contenuto[k];
        marker--;
        return retval;
    }
}
```

```
public static void main(String args[]) {
    try {
        Pila s=null;
        int type=0;
        do {
            try {
                type =Integer.parseInt(
                    JOptionPane.showInputDialog(
                        "Pila (1) o Coda (2)?"));
            } catch (Exception e) {type=0;}
        } while (type<1 || type>2);
        switch (type) {
            case 1: s=new Pila(); break;
            case 2: s=new Coda(); break;
        }
    }
}
```

Usare
Pile e
Code

Polimorfismo!

```

    for (int k=0;k<10;k++)
        if (k%2!=0)
            s.inserisci(new Integer(k));
        else
            s.inserisci(new Float(k*Math.PI));
    for (int k=0;k<11+1;k++)
        System.out.println(s.estrai());
} catch (AssertionError a){
    a.printStackTrace();
} finally {
    System.exit(0);
}

```

Polimorfismo!

Usare
Pile e
CodeDynamic
binding

Modificatori: abstract



Classi dichiarate abstract non possono essere istanziate, e devono essere subclassate.

Metodi dichiarati abstract devono essere sovrascritti

Una class non abstract non può contenere abstract methods



Fondamenti di Java

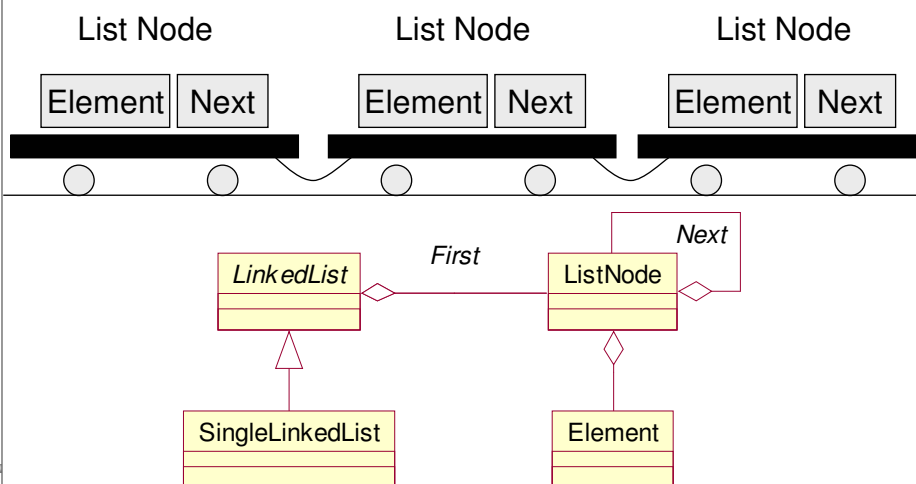
Fac.Scienze - Università di Trento

UML



UML - Linked List

Fac.Scienze - Università di Trento





UML – Double Linked List

Fac.Scienze – Università di Trento

