## Programmazione Avanzata

## 16 Luglio 2019

Rispondere alle domande nello spazio disponibile. Se questo non fosse sufficiente continuare nel retro della pagina.

		Nome e Cognome:
		Matricola:
		Parola Chiave:
1	1.	Descrivere le differenze tra $type$ -inference e $type$ -checking indicando quali dei due approcci vengono usati nei linguaggi $F\#$ e $Java$ .

2. De	
Qu	ale ruolo giocano <i>tipo statico</i> e <i>tipo dinamico</i> nella selezione di un metodo da inv
Qu	ale ruolo giocano <i>tipo statico</i> e <i>tipo dinamico</i> nella selezione di un metodo da inv
Qu	ale ruolo giocano <i>tipo statico</i> e <i>tipo dinamico</i> nella selezione di un metodo da inv
Qu	ale ruolo giocano tipo statico e tipo dinamico nella selezione di un metodo da inv
Qu	ale ruolo giocano tipo statico e tipo dinamico nella selezione di un metodo da inv
Qu	ale ruolo giocano tipo statico e tipo dinamico nella selezione di un metodo da inv
Qu	tale ruolo giocano tipo statico e tipo dinamico nella selezione di un metodo da inv
Qu	ale ruolo giocano tipo statico e tipo dinamico nella selezione di un metodo da inv
Qu	tale ruolo giocano tipo statico e tipo dinamico nella selezione di un metodo da inv
Qu	ale ruolo giocano tipo statico e tipo dinamico nella selezione di un metodo da inv
Qu	ale ruolo giocano tipo statico e tipo dinamico nella selezione di un metodo da inv
Qu	ale ruolo giocano tipo statico e tipo dinamico nella selezione di un metodo da inv

1 3. Considerare il seguente classi Java:

```
class ClasseA {
        public void m1( ) {
                 System.out.println("ClasseA->m1()");
                 m2();
         }
        public void m2( ) {
                 System.out.println("ClasseA->m2()");
         }
}
class ClasseB extends ClasseA {
         public void m2( ) {
                 System.out.println("ClasseB->m2()");
        }
}
class ClasseC {
        public void m( ClasseA a )
                 System.out.println("ClasseC->m()");
        }
}
class ClasseD extends ClasseC {
        public void m( ClasseB a )
                 System.out.println("ClasseD->m()");
        }
}
Descrivere (motivando la risposta) il risultato dell'esecuzione della seguente porzione di
```

Descrivere (motivando la risposta) il risultato dell'esecuzione della seguente porzione di codice:

```
ClasseA a = new ClasseB();
a.m1();
ClasseC d = new ClasseD();
d.m(a);
```

16 Luglio 2019

Programmazione Avanzata

 $(\underline{\text{Pag. }4/8)}$ 

2 4. Consideriamo le seguenti classi:

```
public class Rettangolo {
        private final double base;
        private final double altezza;
        public Rettangolo (double base, double altezza) {
                 this.base = base;
                 this.altezza = altezza;
        }
        public double getBase() {
                 return base;
        public double getAltezza() {
                 return altezza;
        }
        public double getArea() {
                 return base*altezza;
        }
        public double getPerimetro() {
                 return 2*(base+altezza);
        }
}
public class Quadrato extends Rettangolo {
        public Quadrato( double lato ) {
                 super(lato, lato);
        }
}
Viene soddisfatto il principio di Liskov? Motivare la risposta
```

16 Luglio 2019

Programmazione Avanzata

(Pag. 6/8)

<ul><li>5.</li></ul>	Definire differenze e similitudini tra stream ed iteratori in Java.

2	6.	Consideriamo le due seguenti porzioni di codice (dove l è un'istanza della classe Lock): syncrhonized (1) {
		l.lock(); //Blocco di Codice l.unclock();
		Possiamo considerarle equivalenti?