Programmazione Avanzata

11 Giugno 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. Consideriamo la seguente definizione di funzione in F#:

```
let myFunction a =
    if (a>0) then 1.0
    else -1.0
```

Assumiamo, inoltre, che la seguente espressione/assegnamento venga valutata correttamente senza alcun tipo di errore:

```
let x = List.map myFunction lst
```

Quale tipo deve avere la variabile `lst`:

_____

_____

_____

Quale tipo verà assegnato alla variabile `x`:

_____

_____

_____

2 2. Consideriamo la seguente dichiarazione di una interfaccia in Java:

```
interface InterfaceA {

        public void aMethod( List<Integer> l );

        public void aMethod( List<Double> l );

}
```

- ◯ La dichiarazione sarà correttamente compilata.
- ◯ La dichiarazione **non** sarà correttamente compilata..

Motivare la risposta:

1 3. Descrivere il ruolo dei metodi equals, toString e hashcode definiti nella classe Object ed il loro *contratto* d'uso.

2 4. Descrivere il **Principio di Sostituzione di Liskov** fornendo un esempio di classi che lo violano.

1 5. Consideriamo le variabili Java f e g definite come segue:

```
Function<Integer,Integer> f
Function<? super Integer,? extends Integer> g
```

Quali differenze di uso si hanno per gli oggetti referenziati da f e g?

1 6. Consideriamo i seguenti riferimenti a metodi:

1. AClass::instanceMethod
2. AClass::staticMethod
3. obj :: instanceMethod
4. AClass::new

Descriverne significato ed uso (instanceMethod indica un metodo di istanza, mentre staticMethod indica un metodo statico):

2 7. Consideriamo la seguente classe Java che implementa le primitive *test-and-set* di una variabile condivisa:

```
public class CountDown {

        private int counter;

        public CountDown( int init ) {
                this.counter = init;
        }

        public synchronized void go( )
            throws InterruptedException {
                while (this.counter != 0) {
                        this.wait();
                }
        }

        public synchronized void tick() {
                if (this.counter > 0) {
                        this.counter--;
                }
                if (this.counter == 0) {
                        this.notifyAll();
                }
        }

}
```

Quali modifiche dovremmo fare sul codice per rimpiazzare l'istruzione this.notifyAll() con this.notify()?