

# Linguaggi e Programmazione Orientata agli Oggetti

## Prova scritta parziale

16 febbraio 2022, a.a. 2021/2022

1. (a) Per ogni stringa elencata sotto stabilire, motivando la risposta, se appartiene alla seguente espressione regolare e, in caso affermativo, indicare il gruppo di appartenenza (escludendo il gruppo 0).

*Nota bene:* la notazione  $(?: \dots)$  permette di usare le parentesi in un'espressione regolare **senza** definire un nuovo gruppo.

$(0[bB](?:[0-1]^*|[0-1]^+[1L]?)|(\backslash s+)|([a-z]+(?:\backslash.[a-zA-Z\backslash d]^*)^*))$

- i. "0B"
- ii. "0bL"
- iii. "a.b"
- iv. "a."
- v. "a.A3..."
- vi. "aA."

- (b) Mostrare che la seguente grammatica è ambigua.

Exp ::= Exp && Exp | Not  
Not ::= false | true | ! Not | ( Exp )

- (c) Modificare la grammatica definita al punto precedente in modo che **non sia ambigua** e che il linguaggio generato a partire dal non terminale Exp **resti invariato**.

2. Sia `select` :  $( 'a \rightarrow 'a \rightarrow \text{bool} ) \rightarrow 'a \text{ list} \rightarrow 'a \text{ list} \rightarrow 'a \text{ list}$  la funzione tale che

`select p [a1;...;an] [b1;...;bn] = [c1;...;cn]`

dove per ogni  $i = 1..n$ ,  $c_i = a_i$  se  $p\ a_i\ b_i$  è `true`,  $c_i = b_i$  altrimenti; se  $l_1$  e  $l_2$  non hanno la stessa lunghezza, allora `select p l1 l2` solleva un'eccezione (usare `raise (Invalid_argument "select")`).

Esempi:

```
select ( < ) [0;2;5] [1;3;4]=[0;2;4];;  
select ( < ) [] []=[];;  
select ( < ) [0] [1;3;5] solleva un'eccezione  
select ( < ) [0;2;4] [1] solleva un'eccezione
```

- (a) Implementare `select` senza uso di parametri di accumulazione.
- (b) Implementare `select` usando un parametro di accumulazione affinché la ricorsione sia di coda.

3. Considerare le seguenti dichiarazioni di classi Java:

```

public class P {
    String m(float f1, float f2) {return "P.m(float,float)";}
    String m(double[] d) {return "P.m(double[])";}
}
public class H extends P {
    String m(float f1, float f2) {return "H.m(float,float)";}
    String m(double d1, double d2) {return "H.m(double,double)";}
}
public class Test {
    public static void main(String[] args) {
        P p = new P();
        H h = new H();
        P p2 = h;
        System.out.println(...);
    }
}

```

Dire, per ognuno dei casi elencati sotto, che cosa succede sostituendo al posto dei puntini nella classe `Test` il codice indicato, assumendo che tutte le classi siano dichiarate nello stesso package.

Per ogni caso fornire due o tre righe di spiegazione così strutturate: se c'è un errore in fase di compilazione, specificare esattamente quale; se invece la compilazione va a buon fine spiegare brevemente perché e descrivere cosa avviene al momento dell'esecuzione, anche qui spiegando brevemente perché.

- (a) `p.m(42L, 42L)`
- (b) `p2.m(42L, 42L)`
- (c) `h.m(42L, 42L)`
- (d) `p.m(42.0, 42.0)`
- (e) `p2.m(42.0, 42.0)`
- (f) `h.m(42.0, 42.0)`