Linguaggi di Programmazione (III Anno), Luglio 1998 - OOP in Java

Esercizio OOP.1

```
print(p2.m(p1)); 0
print(p2.m(p2)); 0
print(h.m(p1)); 0
print(h.m(h)); 2

print(p1.n(h)); 4
print(p2.n(h)); 5
print(h.n(h)); 5
```

si deve tener presente come si risolve l'overloading (tipo statico degli argomenti), e l'overriding.

Esercizio OOP.2

```
class point extends set_of_points {
private real x';
public point(real x) { x'=x; }
public real x() {return x'; }
public move(real d) { x'=x'+d; }
public scale(real s) { x'=x'*s; }
public bool in (point p) { return(p.x==x'); }
}
class shifted_set extends set_of_points {
private real x';
private list_of_sets l';
public shifted_set(real x, list_of_sets l) { x'=x; l'=l; }
public move(real d) { x'=x'+d; }
public scale(real s) { x'=x'*s; ``for u in l' do u.scale(s)''; }
public bool in (point p)
     { point p'=point(p.x-x');
       return(``exists u in l' s.t. u.in(p')''); }
}
```

Linguaggi di Programmazione (III Anno), Luglio 1998 - FP

Esercizio FP.1

1. $\lambda f, x.cons(x, f\ x)$ ha tipo piu' generale $(X \to X\ list) \to X \to X\ list$
2. $\lambda f, x.cons(f\ x, x)$ ha tipo piu' generale $(X\ list \to X) \to X\ list \to X\ list$

Esercizio FP.2

```
fun shift d (point {x=a}) = point {x=a+d}
  | shift d (shifted_set {x=a, L=B}) = shifted_set {x=a+d, L=B});
fun scale s (point {x=a}) = point {x=a*s}
  | scale s (shifted_set {x=a, L=B}) = shifted_set {x=a*s, L=map (scale s) B});
```

Esercizio FP.3

- Il tipo piu' generale di `btree_it` e' $(X * Y * Y \to Y) \to Y \to X\ btree \to Y$
- la istanziazione di `btree_it` che calcola il numero di nodi di un albero e'

  ```
  val numero_nodi = btree_it (fn (_,n1,n2) => n1+n2+1) 0;
  ```

- la istanziazione di `btree_it` che calcola la lista dei nodi secondo la visita inorder e'

  ```
  val lista_nodi_inorder = btree_it (fn (x,l1,l2) => l1@(x::l2)) nil;
  ```