Final Examination Answer Sheet

TIME: 06/16/2009, 14:20–17:20

Feel free to as the TAs for additional answer sheets when necessary. But please mark your names clearly and staple your sheets together.

Your NTU ID Number:

Your Chinese Name:

Anything You Want to Say about the Class (optional):

1 POO Zoo

(1)	basic answer: assume that the first part can be compiled, just new three classes without the inheritance hierarchy, and use several lines for action. ideal answer (with bonus 2 points): say that the first part cannot be compiled. Use whateverway you can do (say, println) to print the messages
(2)	• Let the three classes extends `Unicorn` • Move `do_unicorn_walk` to class `Unicorn`, and merge it with the original `walk`.

2 POO Calculus

(1)

```
class Exp extends Function implements Differentiable, Integrable{
    public double eval(double x){ return Math.exp(x); }
    public Function diff(){ return this; }
    public Function inte(){ return this; }
}
```

(2)

```
class Poly extends Function implements Differentiable, Integrable{
    double deg;
    double coef;
    Poly(double coef, double degree){ this.coef = coef; this.deg = deg; }
    public double eval(double x){ return coef * Math.pow(x, deg); }
    public Function diff(){ return new Poly(coef * deg, deg-1); }
    public Function inte(){ return new Poly(coef / (deg+1), deg+1); }
}
```

Bonus up to 5 points if doing some special case handling (for instance, `deg = -1` in `inte`.)

(3)

```
class Sum extends Function implements Differentiable, Integrable{
    Function f, g;
    Sum(Function f, Function g){ this.f = f; this.g = g; }
    public double eval(double x){ return f.eval(x) + g.eval(x); }
    public Function diff(){ return new Sum(
      ((Differentiable)f).diff(),
      ((Differentiable)g).diff()); }
    public Function inte(){ return new Sum(
      ((Integrable)f).inte(),
      ((Integrable)g).inte()); }
}
```

A class casting exception would be thrown when calling the corresponding function if `f` or `g` is not both differentiable and integrable.

(4)

```
        System.out.println(
                          new Sum(
                                  new Sum(new Poly(3, 2),
                                          new Poly(2, 1)),
                                  new Sum(new Poly(5, 0),
                                          new Exp().inte())
                                  ).diff().eval(2)
                          );
```

Feel free to ask the TAs for additional answer sheets when necessary. But please mark your names clearly and staple your sheets together.

Your NTU ID Number: ______________________

Your Chinese Name: ______________________

3 POO Search

(1)	`class Searcher<E extends Searchable> extends ArrayList<E>`
(2)	`Searcher<E> search(String query)`
(3)	`Searcher<E> addSearchResult(Searcher<? extends E> orig_list , String query)`

4 POO Exception

<table>
<tr>
<td>(1)</td>
<td><pre>
boolean action_0_flag = true;
try{
  action_1();
  if (action_2()) {action_3(); action_0_flag = false; }
  else return;
}
catch(NullPointerException e1){
  action_4(); action_5(); action_6();
}
catch(NetworkdisconnectException e2){
  action_4(); action_6(); action_7();
}
catch(UserNotExistException e3){
  action_4(); action_6(); action_8();
}
catch(Exception e4){
  action_4(); action_6();
}
catch(Throwable e5){
  action_4();
}
finally{
  if (action_0_flag) action_0();
}
</pre></td>
</tr>
</table>

5 POO Love

(1)	Yes.
	`love_count = 0; //initially`
	`r = love_count + 1; //from J1, r = 1`
	`r = love_count + 1; //from J2, r = 1`
	`r = love_count + 1; //from J3, r = 1`
	`love_count = r; //from J1, now love_count = 1`
	`love_count = r; //from J2, now love_count = 1`
	`love_count = r; //from J3, now love_count = 1`

(2)	Yes. Because each thread holds a different lock (the thread itself), and hence still works asynchronously.
(3)	No. Because each thread shares the same lock (the user M object), and hence should work synchronously.

(4)	`synchronized(receiver){`
	`  increase_love_count(receiver);`
	`}`