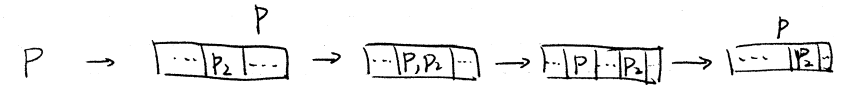
CST Part IA: OOP, SV 1
Joe Yan
2016-11-6

1 Example Sheet 1314 3B

```
public static float[][] creatMatrix(int n) {
        float[][] mat = new float[n][n];
        return mat;
}
public static float[][] transpose(float[][] mat) {
        float store;
        int len = mat.length;    for(int i = 0; i < len; ++i) {
                for(int j = i; j < len; ++j) {
                        store = mat[i][j];
                        mat[i][j] = mat[j][i];
                        mat[j][i] = store;
                }
        }
        return mat;
}
```

2 Example Sheet 1314 6A



3 Example Sheet 1314 11B

1.
```
public class Vector2D {
          public float a;
          public float b;

          public Vector2D() {
                  a=0;
                  b=0;
          }
          public Vector2D(float x,float y) {
                  a = x;
                  b = y;
          }

          public static float scalarProduct(Vector2D x, Vector2D y) {
                  float t = x.a*y.a+x.b*y.b;
                  return t;
          }

          public static Vector2D normalise(Vector2D x) {
                  float mag = magnitude(x);
                  return new Vector2D(x.a/mag,x.b/mag);
          }

          public static float magnitude(Vector2D x) {
                  float t = (float) Math.sqrt(x.a*x.a+x.b*x.b);
                  return t;
          }

          public void add(Vector2D v) {
                  this.a = this.a + v.a;
                  this.b = this.b + v.b;
          }
}
```

2. - Make fields private
   - Be careful when the reference is passed in or out, when construct the object or provide the specific field, always make a new reference for it. (Not for this example)
   - Final the class or method preventing it from overriding
   - Do not provide any methods can mutate the field

3. Mutable:

```
public void add(Vector2D v) {
                this.a = this.a + v.a;
                this.b = this.b + v.b;
        }
```

   Immutable:

```
public Vector2D add1(Vector2D v) {
        return new Vector2D(this.a+v.a,this.b+v.b);
}
public Vector2D add(Vector2D v1, Vector2D v2) {
        return new Vector2D(v1.a+v2.a,v1.b+v2.b);
}
public static Vector2D add1(Vector2D v1, Vector2D v2) {
        return new Vector2D(v1.a+v2.a,v1.b+v2.b);
}
```

4. Declare in annotation?

I do not quite understand what is 3 and 4 asking for..

4 Example Sheet 1314 12B

```
public class OOPLinkedList {
        protected OOPLinkedListElement hd;
        public OOPLinkedList() {
                hd = null;
        }

        public void add(int x) {
                hd = new OOPLinkedListElement(x,hd);
        }
        public void remove() {
                if(hd == null) hd = null;
                else hd = hd.getnext();
        }
        public int get() throws Exception {
                if(hd==null)throw new Exception(”List is null.”);
                else return hd.getele();
        }

        public int length() {
                if(hd == null) return 0;
                int len = 1;
                OOPLinkedListElement p = hd;
                while(p.getnext() != null){
                        ++len;
                        p=p.getnext();
                }
                return len;
        }
}

public class OOPLinkedListElement {
        private int ele;
        protected OOPLinkedListElement next;

        public OOPLinkedListElement(int x) {
                ele = x;
                next = null;
        }

        public OOPLinkedListElement(int x,OOPLinkedListElement xs) {
                ele = x;
                next = xs;
        }

        public int getele() {
                return ele;
        }

        public OOPLinkedListElement getnext() {
                return next;
        }
}
```

5 Example Sheet 1314 18B

```
public class OOPSortedLinkedList extends OOPLinkedList{

        public OOPSortedLinkedList() {
                super();
        }

        public void add(int x) {
                if(hd == null) {
                        hd = new OOPLinkedListElement(x,null);
                        return;
                }
                if(x<hd.getele()) {
                        hd = new OOPLinkedListElement(x,hd.getnext());
                        return;
                }
                OOPLinkedListElement before = hd;
                while(before.getnext() != null && x > before.getnext().getele()) {
                        before = before.getnext();
                }
                before.next = new OOPLinkedListElement(x,before.getnext());
        }
}
```