Designing a List in Java

SOftEng
http://softeng.polito.it

Assignment

- Write a list of integers using C.
  - Insert: insert at head
  - First: return the first element
  - Remove: remove the first element
  - Size: return number of elements

  - Do not consider error management.

Usage example

```
#include "list.h"
int main(int argc, char* argv[]){
  init();
  insert(1);
  insert(2);
  insert(3);
  printf("Size: %d\n",size());
  printf("First: %d\n",first());
  delete();
  printf("Removed\n");
  printf("First: %d\n",first());
}
```


Solution

```
typedef struct el {
  struct el* next;
  int info;
} element;
element* head;
void init(){
  head = NULL;
}
void delete() {
  head = head->next;
}
int first(){
  return head->info;
}
```

```
void insert(int i){
  element* p=(element*)malloc…
  p->next = head;
  p->info = i;
  head = p;
}
int size(){
  int count=0;
  element *p=head;
  for(;p!=NULL;p=p->next)
     count++;
  return count;
}
```

A couple of problems

- How can we use two distinct lists in the same program?
  - Copy and paste is not scalable
- How to ensure a proper initialization?
  - No one forces the user to call init()

Sample usage of List

```
public static void main(String[] Args){
  List list = new List();
  list.insert(1);
  list.insert(2);
  list.insert(3);
  System.out.println("Size: " + list.size());
  System.out.println("First: "+ list.first());
  list.remove();
  System.out.println("Removed");
  System.out.println("First: "+ list.first());
}
```

Design Process

- Identify main concepts concepts
  - Classes
- Identify relationships among concepts
  - Associations
- Identify the information stored in objects
  - Attributes
- Identify operations performed on objects
  - Methods

List design