

CS360 Lecture 8

Polymorphism

Thursday, February 26, 2004

Overloading and Overriding:

I have been using both these terms throughout this semester. What is the difference between them?

- Override a method when a method in a subclass has exactly the same signature as an inherited method.
- Overload a method when they have the same name but different signature

Constructor Chaining

What is the difference between the following two statements:

- `super(x, y);`
- `this(x, y);`

Polymorphism

```
A a = new B(5);  
a.m();
```

Both A and B contain a method m(). Which method will get called?

The real type for a is B while the declared type for a is A.

Advantages of Polymorphism

One of the advantages is that you can avoid a program with lots of tests (if, switch statements).

```
for( int i=0; i<size; i++ )  
    if( figures[i] instanceof Rectangle )  
        //draw a rectangle  
    else if( figures[i] instanceof Circle )  
        //draw a rectangle
```

Polymorphism is a fundamental issue in OOP and the basis for an efficient use of inheritance.

Polymorphism is achieved by the late binding mechanism. The method called is determined at run-time not compile-time.

Problem:

Design and implement an application for creating email addressbooks. An email addressbook will contain entries that would either be an individual entry or a group entry. An individual entry will have a nickname, firstname, lastname and email address. A group entry will only have a nickname, and a list of all the members of the group.

The first step of the design is to create the inheritance hierarchy. You need to determine the classes, interfaces and abstract classes you will need for this program.

The only specifications that your boss requires is that you create an AddressBook class.

An addressbook can be created by specifying the maximum number of entries it should have, or by specifying nothing, in which case a default of 10 entries will be used.

e.g.,

```
myaddbook = new AddressBook();  
youraddbook = new AddressBook(20);
```

Methods in the addressbook class:

All the *boolean* methods return a true value if the operation was successful, and a false value if it was not.

1. `boolean addNewContact(String nick, String lastname, String firstname, String email)`

Adds a new contact in the addressbook. This method must check for the following error situations:

- The addressbook is full - in which case it should simply generate an error message but should not crash.
- A contact already exists with the given nick - again it should generate an error message and show the contact with the given nick.

2. `boolean addNewGroup(String nick)`

Adds a new Group in the addressbook with the given nick. This should also check for the error situations above. A group nick cannot be the same as an existing contact nick.

3. `boolean addContactInGroup(String groupnick, String contactnick)`

Adds the nickname of the contact as a member of the Group with the nick "groupnick"

This should check for the following conditions to be satisfied:

- A group already exists with the given groupnickname
- A contact already exists with the given contactnickname

If either of the above is not satisfied, then the method should generate an error (as usual, it should not crash).

4. `boolean addContactInGroup(String groupnick, String contactnick, String firstname, String lastname, String email)`

Like the above, but creates a new contact before adding the contact to the group. You will need to use other appropriate methods in this class to achieve this, and check for success of that operation as above.

5. `boolean deleteEntry(String nick)`

Deletes the entry with the given nickname for an individual entry, this deletes only the individual contact. For a group entry, this deletes the group entry, as well as all the contacts that belong to the group. (Note that either way, if an entry is deleted, if this entry belongs to a group, then that entry should be removed from the group in question as well to avoid hanging references.

6. `void displayEntry(String nick)`

Will display the given entry on System.out this will display the information about the Entry with the given nickname on the screen. If the nickname represents a regular contact this will show the nickname, full name, and email address. If it is a group, it should show the information on all the entries in the group. Should show an error if the entry is not present.

7. `void displayAll()`

Will display all entries of the addressbook on System.out.