

LECTURE NOTES OF WEEK 7

25/03/2008--27/03/2008

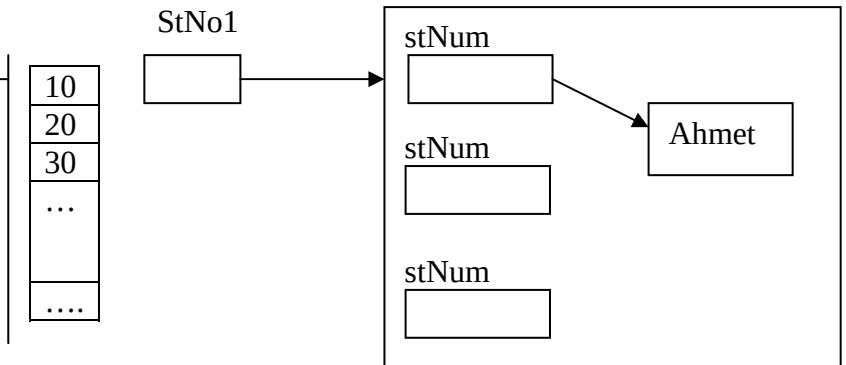
Note Takers:

- Fatma Ayhan
- Mehmet Fatih Bolukbasi
- Özge Vargel

☼ 25.03.2008 Tuesday

```
Student [] strArray= new Student[100];
Int [] intArray= new int [100];
Student stNum=new Student();
StuNo1 stNo=10;
```

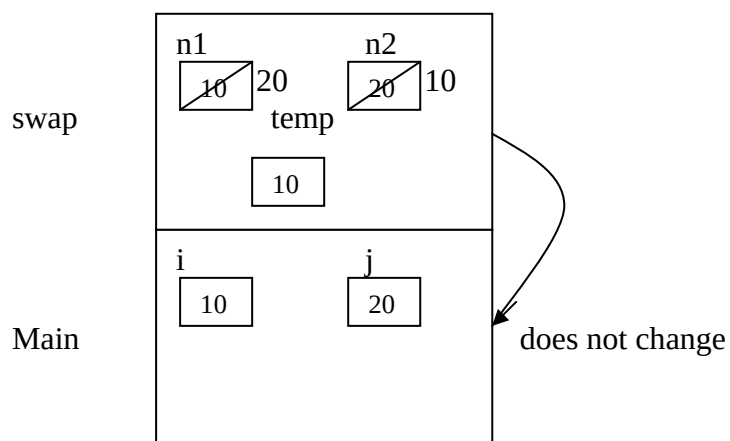
Class=> user defined data type



Call by Value

```
int i =10;
int j =20;
swap ( i, j);
System.out.print(i, j);
```

```
void swap(int n1, int n2) {
    int temp=n1;
    n1=n2;
    n2=temp;
}
```

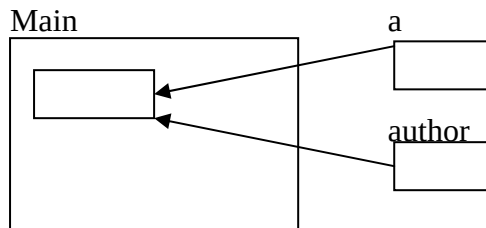


```

public class Book {
    private String author, title, type;
    private int noOfCopies;
    public Book(String a, String t, String ty) {
        author= new String(a); //author=a;
        title= new String(t); //title=t;
        type= new String(y); //type=y;
        noOfCopies=1;
    }
}

```

Book nBook= new Book ("Yasar Kemal", "TM", "novel")



*Instructors note: "getter methods are accessors, setter methods are mutators."

equals()

Default implementation in object class

```

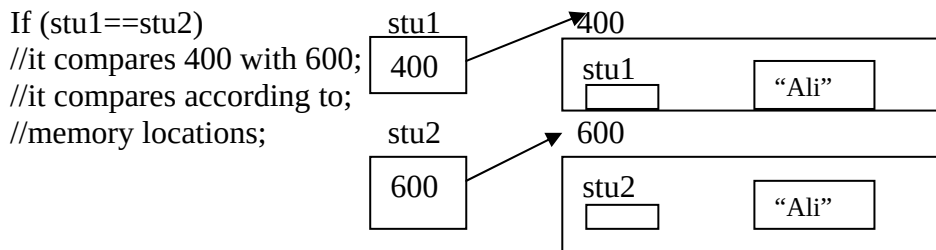
Public boolean equals(object obj) {
    Return (this==obj);
}

```

```

Student st1= new Student ("Ali");
Student st2= new Student ("Ali");

```



```

//check quality of books;
public boolean equals(object b) {
    Book bk= (Book) b;
    Return (this.author.equalsIgnoreCase(bk.author) &&
        title.equalsIgnoreCase(bk.title) &&
        type.equalsIgnoreCase(bk.type));
}

```

Type casting

```

getAuthor()
getCount()
getTitle()

```

```
public void increaseBookCount() {
    noOfCopies++;
    toString();
}
```

⌘ 27.03.2008 Thursday

Assignment No 1__due date: April 8, Tuesday

ArrayList

A member of java.util library.

java.util.ArrayList
*ArrayList() *add(); ...

Book class→already defined

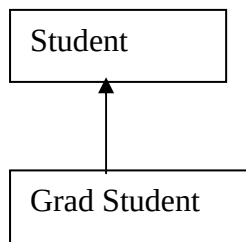
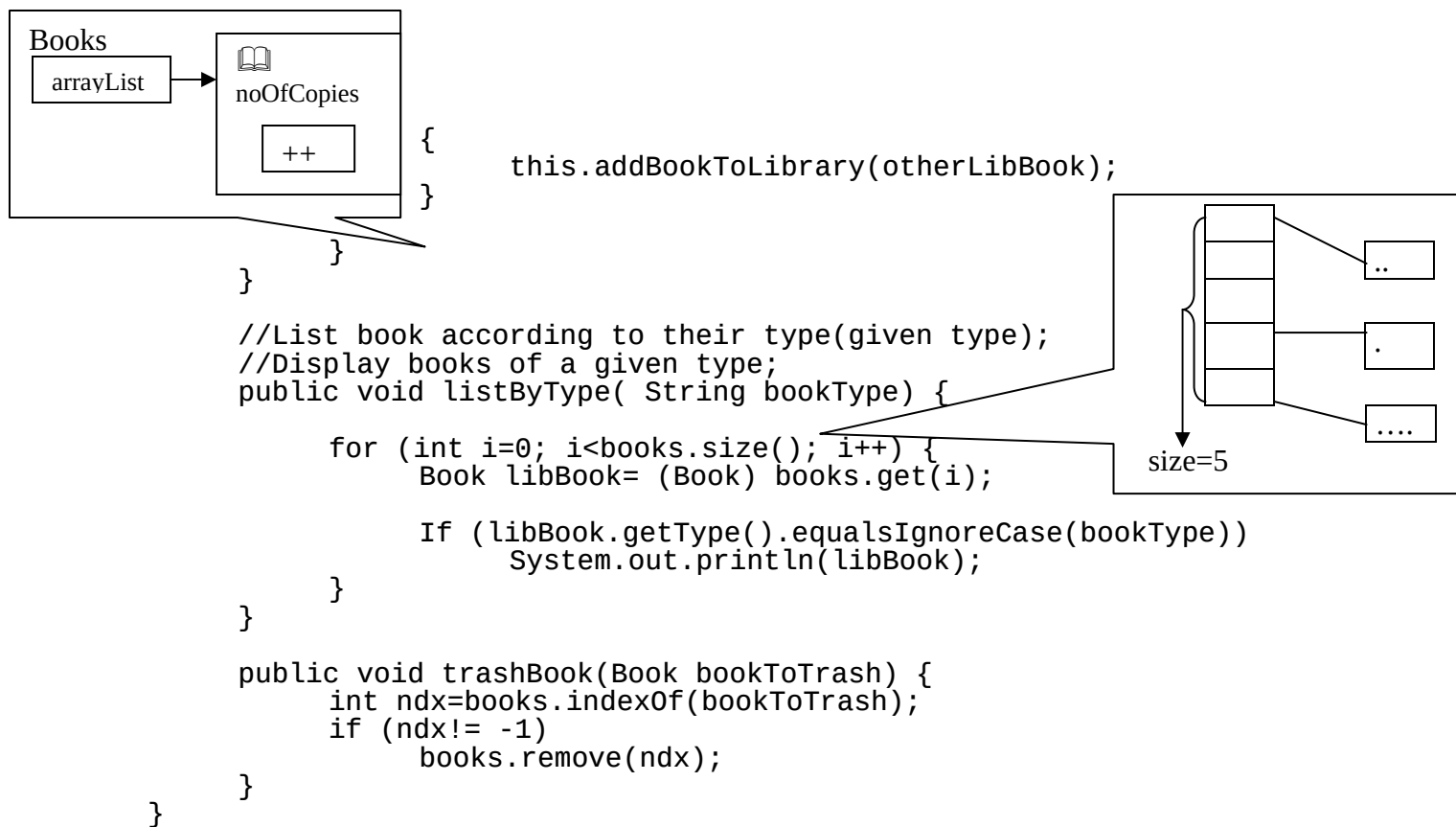
main

```
{
privateLibrary yourLibrary= new privateLibrary();
yourLibrary.addBookToLibrary(book1);
...
...
...
}
```

```
public class privateLibrary {
    private ArrayList books;
    public privateLibrary() { books= new ArrayList(); }
    public void addBookToLibrary(Book bookToAdd) {
        int ndx= books.indexOf(bookToAdd);
        if (ndx== -1)
            books.add(bookToAdd);
        else
        {
            //we have some copies of bookToAdd
            Book libBook= (Book) books.get(ndx);
            libBook.increaseBookCount();
        }
    }
    //Add all books of another library to this library;
    public void addLibrary(privateLibrary otherLibrary) {
        for (int i=0; i<otherLibrary.books.size(); i++) {
            Book otherLibBook= (Book)otherLibrary.books.get(i);
            //How many copies do we have in otherLibBook?;
            for(int ncpy=1;ncpy<=otherLibBook.getCount();ncpy++)
```

Type
casting

Book class:
Public void increaseBookCount()
{
noOfCopies++;
}



```

Student stu2=Gradstu1 ☑
GradStu grad1= stu2 ☒

```

1) Generally, we use type casting when appending in an arrayList. What is the main reason for this application?

Unless we use type casting, a compilation error about incompatible data types can be occurred.

2) What is the advantage of using arrayList instead of an array?

The size of the array is fixed but an array can be extended by adding new elements to it. Thus, it is not necessary to know or declare the size of it before.

3) What are the main plausible applications by using the methods of arrayList class?

append:
arrayListName.add(object);

get an object:
dataType identifier= (DataType) arrayListName.get(index);

omiting an object from the list:
arrayListName.remove(int index);

Adding a new element to a specific index;
`arrayListName.add(int index, object);`

Removing all elements;
`arrayListName.clear();`

Updating an element :
`arrayListName.set(index, object);`

Returning size:
`arrayListName.size();`

* Note that in order to use these methods, "java.util.ArrayList" should be imported.

4) How can we call the methods which are implemented in arrayList class?

`arrayListName.methodName();`