IST 256
Lab Week 13, Monday, November 22, 2010

1. Extending the program with Student data from a file

In this example, we will again extend the Students example that we have been working on in lab. Today we will add dorm address data for each student, and we will add a button to the form so that the user can find the tallest student.

a. Start by **opening the Students** project.

b. Create additional data.

Edit the file students.txt to have two additional items of data for each student. Add a dorm name and a dorm room number, separated by commas, on each line. Each line should look something like:

```
Alex,M,20,73,Dellplain Hall,326
```

Don't forget that there shouldn't be any extra blanks around the commas.

c. Create a DormAddress class:

We are going to represent the dorm name and room together in a class for dorm addresses. In the left pane of NetBeans, find the Students project and right click on the name Students. In the menu, select New -> Java class. In the new class window:
give the **class name** as DormAddress
select the **package** of the class to be students

d. Write the code for the DormAddress class.

Write a class:

- Put two fields in the class, both are Strings representing the dorm name and room.
- Put a constructor that can initialize both fields.
- Put a method called toString that returns a String with the room followed by the dorm name, and labeled as the dorm address, e.g.
  - Dorm address: 326 Dellplain Hall

e. Extend the Student class.

In the code for the Student class:

- Add a field to keep a dorm address.
- Check that there is an accessor method called getHeight to return the height of the student (our GUI program is going to use this to find the tallest student).
- Check that there is an accessor method to return the name of the student (to show the name of the tallest student with displaying all the student information.)

- Change the toString method so that the dorm address is added to the end of each student's string. For this, you can call the DormAddress toString method.

f. Extend the form interface.

On the form, we will keep the previous buttons and labels for reading the file and saving the data, and for displaying, sorting and searching for students. Now we are going to add a button and a label so that we can find and display the tallest student. The form can look something like this:

```
|__ Read Student File__|      File Status

|___Display Students___|                        |__Find Tallest Student__|
        Results:                                      Tallest Student:

(other form elements . . .)
```

For the new button, select Event -> action -> actionPerformed.

g. Revise the code for the GUI.

Check the students array and variable numstudents at the top of the program and remember what variable names you used if they are different.

In the first button to read the file:

- With the other variable declarations, add the declarations for two variables to read from the file for the dorm name and room number.
- Inside the while loop, add two more if statements to read these two lines from the file. Which hasNext and next methods should you use?
- Change the call to the Student constructor to pass the dorm name and room.

No changes need to be made to the second button to display the students, except that the toString method returns longer strings as the dorm information has been added and you may need to adjust the size of the form.

h. Test your program so far and make sure that the dorm information is displaying correctly.

When you click on the Display students button, does the room number and dorm name show up for each student?

i. Write the code for the button to find the tallest student.

As discussed in the slides for searching, this problem doesn't try to find the maximum value in an array, but the index of the student with the maximum height value. This means that in the search algorithm, we will save the array index of the student with the maximum height found so far.

The core of this code will be something like:

```
// start with the index of the first student is the tallest student so far
int maxindex = 0;

// loop over the remaining students
for ( int index = 1; index < numstudents; index++)
{
  // if the next student is taller than the one found so far, make it be the tallest
        if (students[index].getHeight() > students[maxindex].getHeight())
        {
                maxindex = index;
        }
}
// use maxindex to get a student and display their name and height
// display the student name and height in the label for the tallest student
```

Write the variable declarations and the remaining code to finish this button.

j. Test your program.

Which student is displayed as the tallest? Is this the first or last of the tallest students?

Hand in your lab sheet today for Week 13 and have a Happy Thanksgiving!