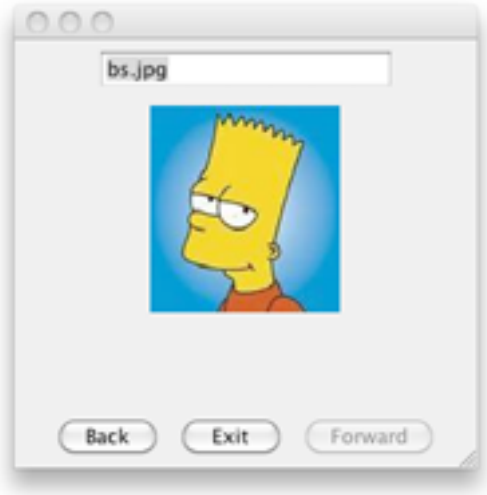
Programming Assignment 13
Due Friday, April 30

Here is how a browser such as Firefox or Chrome uses a stack.

For this program you will simulate the back and forward buttons of a web browser. These buttons are often implemented using a stack.

Rather than display a website you will display an image. That is, rather than entering www.stonehill.edu or some other website, your program will accept the name of an image such as bart.jpg and display the image. There should be a back button, a forward button, and an exit button as well as a text field that accepts the name of the image.



When the user enters an image filename in the text field and hits "Enter" that image is displayed in the frame. Hitting Enter generates an event (ActionEvent). The back button and forward button act as they do in a web browser.

To begin set up your GUI:

- place a text field on a panel and place it in the NORTH
- place a label on a panel and place it CENTER. The label should have an image on it, for example "bs.jpg" (Bart Simpson)
- place three buttons on a panel and place them in the SOUTH.

You will also need two stacks, say a backButtonStack and a forwardButtonStack. These stacks will hold a string representing image files ("homer.jpg").
Make sure to download a Stack class from the class website (other stuff). Any implementation is OK

. The two Stacks are declared as

Stack<String> backButtonStack = new Stack<String>();

Stack<String> forwardButtonStack = new Stack<String>();

(If you use the array implementation you need to specify the size of the stack:
e.g new Stack(100)).

You should also have a global variable (String) called currentPicture that holds the filename of the currently displayed picture.

This should be initialized to some picture when the program begins.
That is, when the program starts currentPicture should have a value such as "homer.jpg" and the picture should be displayed on the label. Do this in the constructor.

Here are the actions :
When a filename (for example, bart.jpg) is entered in the text field

- currentPicture is pushed on the backButtonstack
- make currentPicture what is now the name in the text field
- display image of currentPicture on the label

When the back button is clicked

- get the current filename from the text field
- push it on the forwardButtonStack
- pop a filename off the backButtonStack, place it in the text field, and display the pic in the label

The forward button is implemented similar to the back button. You figure that out.
NOTE:

The buttons need to be enabled and disabled depending on whether a corresponding stack is empty or your program will have errors. If the backButtonStack is empty, the back button should not work, so disable it.

So that the program can be graded easily, everyone should use the same pictures. There are six images on the class website under "Other Stuff." (Three Simpson pics and three Big Bang Theory pics) Copy these into your folder and use them for "data." These files are all jpg files with names hs, ms, bs, leonard, sheldon, and raj. (Sorry Howard, you lose out)

2. Carla's Crazy Car-Park is a tiny parking garage on 46th St NYC capable of holding just eight cars. There is a single entrance/exit. So if a car, C, has to leave the garage and C is not nearest the exit

- all cars blocking C are temporarily moved out of the garage
- C exits
- and the cars that were moved are brought back into the garage in the original order

Write a program that processes a file *cars.txt* such that each line contains

- The license plate number (String)
- And "A" for arrival or "E" for Exit

Here are a few typical lines

WAH7 A
GRT9 A
WAH7 E

Your program should print a log of the operations.

- When a car arrives it should print the message
  WAH7 arrives
- When a car exits it should print
  WAH7 exits and include the number of times the car had been moved
- If a car arrives and the garage id full it should print
  WAH7 arrives. Garage is full

Here is an example.
Suppose that IN THIS EXAMPLE the garage holds just four cars.

Input: cars.txt	Output (to the screen)
ASD A	ASD arrives
BSD A	BSD arrives
CSD A	CSD arrives
ASD E	ASD exits and was moved 0 times
DSD A	DSD arrives
ESD A	ESD arrives
FAD A	FSD arrives. Garage is full
BSD E	BSD exits and was moved 1 time
CSD E	CSD exits and was moved 2 times
	Etc.

Here is some help:

You are going to need a class Car:
A Car object should have two private fields
String license
int count // keeps track of the number of times the car was moved.

Your Car class should also have:

- Getters and setters

- A method
  public void increment()
  that increments the count each time the car is moves

- boolean equals(Object o) – overridden from Object
  that makes two car objects with the same plate equal

- String to String() – returns the license plate

You will also need a separate class Garage
The Garage class should have two **NON-STATIC** methods:

- One to handle arrivals
- Another to handle departure

The main method should be very simple
Make a scanner from the file
Read the file and call the appropriate method to handle arrivals and departures
You must use a try-catch construction for the file.

Most importantly, you will use

- a **stack of Car** to model the garage and
- another temporary **stack of Car** into which you temporarily place blocking cars when a car exits.