What is a Stack?

A stack is a data structure with two essential operations; push and pop.
- push something onto a stack
- pop the top thing off the stack

A stack of plates is the common metaphor.

The part of memory referred to as the stack uses this approach for allocating and releasing memory.
- NOTE: Once allocated and on the memory stack, random access is possible. The "stack" behavior is just at the allocation level.
Stack Frames

Our examples so far have been mostly limited to a single method call.

When expanding to consider a series of method calls (and their completion) we extend our picture to include "frames" on the memory stack.

Each method call places a new frame on the stack which contains:
- The parameters and other local variables used by that method.
- Other information used "internally" which we will not explore in this course).

Method Calls and Parameters and Memory

public static void main(String[] args) {
    Student s1 = new Student("Pat", 18, 987654321);
    int val1 = 5;
    Integer val2 = 25;

    System.out.println(
        s1 + " Value #1:" + val1 + " Value #2:" + val2
    );

    messWithStudent(s1, val1, val2);

    System.out.println(
        s1 + " Value #1:" + val1 + " Value #2:" + val2
    );
}

public static void messWithStudent(
    Student localStudent, 
    int x, Integer y 
) {
    localStudent.useToken();

    localStudent = new Student("Sam", 345363267);
    localStudent.receiveTokens(x+y);
    x=0;
    y++;

    System.out.println(
        "localStudent+" \tValue #1:"+x+" Value #2:"+y 
    );
}

Copyright © 2010-2012 : Evan Golub