

Algorithms & Data Structures I:

2_Stack

Today's Topics

- What a stack is
- Stack applications
- Implementation
 - Using an array
 - Using a linked list
- Keeping minimum in Stack
- Correct brackets sequence

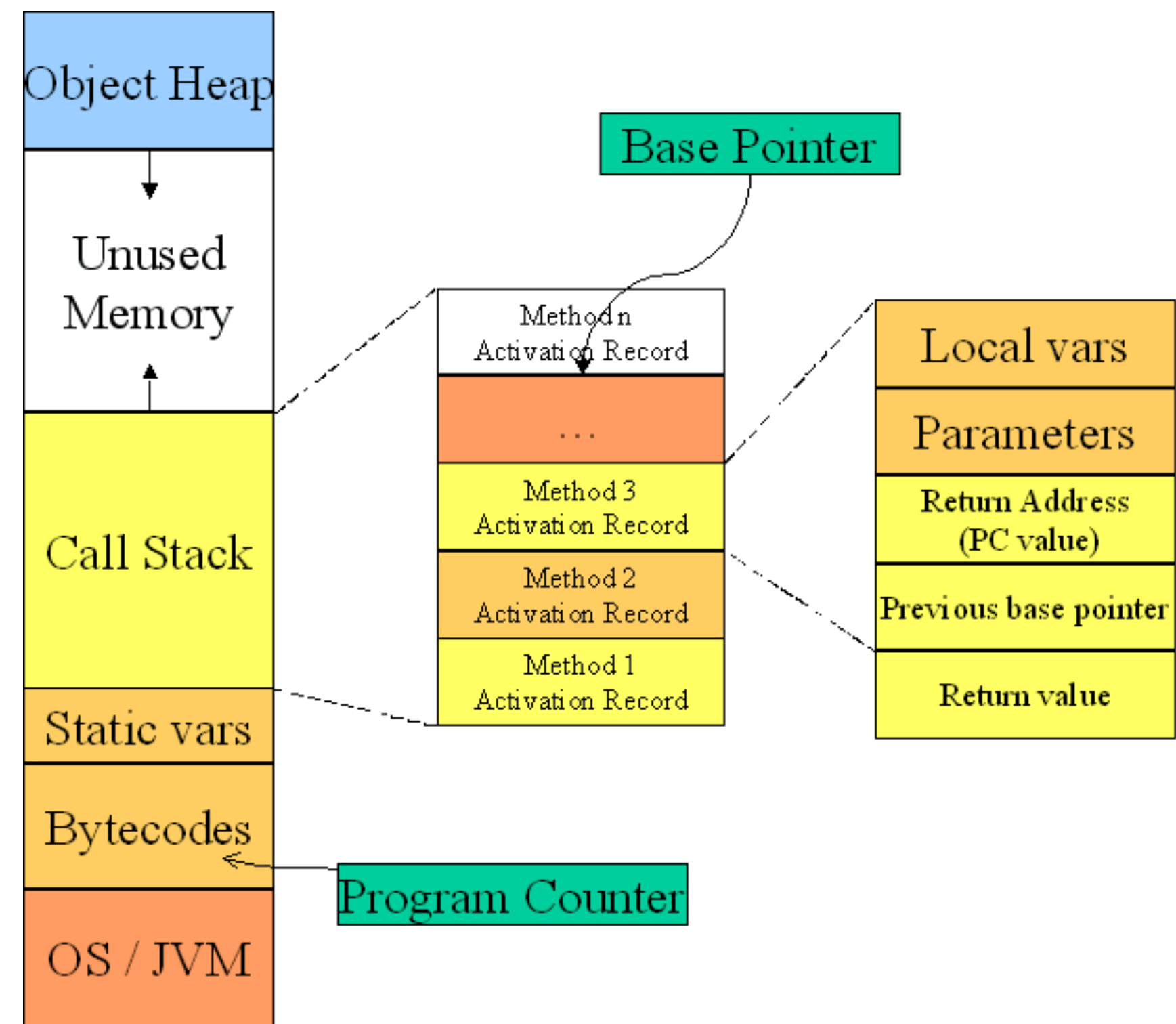
What a stack is

- Push(x)
 - add an element to the top
- Pop()
 - remove last added element from stack and return it
- LIFO
 - **Last In First Out**

Applications

- Operation Systems
 - Memory management
 - Function Call Stack
- Calculators
 - Expression evaluation
 - Parenthesis matching

Call Stack



Array Implementation Idea

Array Implementation.Push

```
void Push(x):  
    if head == capacity:  
        ExtendStack()  
    array[head] = x  
    head = head + 1
```

Array Implementation.Pop

```
type Pop(x):  
    if head == 0:  
        error "underflow"  
    else:  
        head = head - 1  
        return array[head]
```


Array Implementation.Extend

```
void ExtendStack(x):  
    capacity = capacity * 2  
    allocate new_array(capacity)  
    copy array into new_array  
    free(array)  
    array = new_array
```

Linked List Implementation Idea

Linked List Implementation.Push

```
void Push(x):  
    allocate new_head  
    new_head.value = x  
    new_head.next = head  
    head = new_head
```

Linked List Implementation.Pop

```
type Pop(x):  
    if head == null_ptr:  
        error "underflow"  
    else:  
        return_value = head.value  
        tmp_head = head  
        head = head.next  
        free(tmp_head)  
        return return_value
```

Pros & Cons

Keeping minimum in Stack

Correct brackets sequence. Definition

Correct brackets sequence . Example

Correct brackets sequence.Problem

Correct brackets sequence.Algorithm

Correct brackets sequence.Pseudocode

```
bool Checker(x):  
    for symbol in S:  
        if (isClosedBracket(symbol)):  
            if (stack.empty()):  
                return false  
            else:  
                if (classOf(stack.pop()) !=  
                    classOf(symbol)):  
                    return false  
        else:  
            stack.push(symbol)  
    return stack.empty()
```

Your questions!