Practical

Array implemented in stack:

case1

class Stack {

static final int MAX = 12;

int top;

int a[] = new int[MAX];

boolean isEmpty()

{

return (top < 0);

}

Stack()

{

top = -1;

}

boolean push(int x)

{

if (top >= (MAX - 1)) {

System.out.println("Stack Overflow");

return false;

}

else {

a[++top] = x;

System.out.println(x + " pushed into stack");

return true;

}

}

int pop()

{

if (top < 0) {

System.out.println("Stack Underflow");

return 0;

}

else {

int x = a[top--];

return x;

}

}

int peek()

{

if (top < 0) {

System.out.println("Stack Underflow");

return 0;

}

else {

int x = a[top];

return x;

}

}

void print(){

for(int i = top;i>-1;i--){

System.out.print(" "+ a[i]);

}

}

}

class main {

public static void main(String args[])

{

Stack s = new Stack();

s.push(11);

s.push(22);

s.push(33);

System.out.println(s.pop() + " Popped from stack");

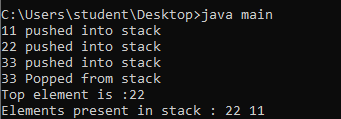
System.out.println("Top element is :" + s.peek());

System.out.print("Elements present in stack :");

s.print();

}

}



Case2

class Stack {

static final int MAX = 2;

int top;

int a[] = new int[MAX];

boolean isEmpty()

{

return (top < 0);

}

Stack()

{

top = -1;

}

boolean push(int x)

{

if (top >= (MAX - 1)) {

System.out.println("Stack Overflow");

return false;

}

else {

a[++top] = x;

System.out.println(x + " pushed into stack");

return true;

}

}

int pop()

{

if (top < 0) {

System.out.println("Stack Underflow");

return 0;

}

else {

int x = a[top--];

return x;

}

}

int peek()

{

if (top < 0) {

System.out.println("Stack Underflow");

return 0;

}

else {

int x = a[top];

return x;

}

}

void print(){

for(int i = top;i>-1;i--){

System.out.print(" "+ a[i]);

}

}

}

class main {

public static void main(String args[])

{

Stack s = new Stack();

s.push(11);

s.push(22);

System.out.println(s.pop() + " Popped from stack");

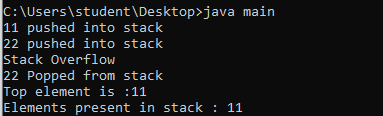
System.out.println("Top element is :" + s.peek());

System.out.print("Elements present in stack :");

s.print();

}

}



Case3

class Stack {

static final int MAX = 2;

int top;

int a[] = new int[MAX];

boolean isEmpty()

{

return (top < 0);

}

Stack()

{

top = -1;

}

boolean push(int x)

{

if (top >= (MAX - 1)) {

System.out.println("Stack Overflow");

return false;

}

else {

a[++top] = x;

System.out.println(x + " pushed into stack");

return true;

}

}

int pop()

{

if (top < 0) {

System.out.println("Stack Underflow");

return 0;

}

else {

int x = a[top--];

return x;

}

}

int peek()

{

if (top < 0) {

System.out.println("Stack Underflow");

return 0;

}

else {

int x = a[top];

return x;

}

}

void print(){

for(int i = top;i>-1;i--){

System.out.print(" "+ a[i]);

}

}

}

class main {

public static void main(String args[])

{

Stack s = new Stack();

System.out.println(s.pop() + " Popped from stack");

System.out.println("Top element is :" + s.peek());

System.out.print("Elements present in stack :");

s.print();

}

}

