**//JAVA ASSIGNMENT**-**6**

**//1. Write a Java program to find the maximum value from the given type of //elements using a generic function.**

import java.util.Scanner;

class TypeCheck<T>

{ T obj[];

int n;

TypeCheck(T obj[], int n)

{

this.obj=obj;

this.n=n;

}

T Maximum()

{

T x;

if(obj.getClass().getName().equals("[Ljava.lang.String;"))

{

int i;

x=obj[0];

for(i=0;i<n;i++)

{

if(((String) obj[i]).compareTo((String)x)>0)

x=obj[i];

}

}

else if(obj.getClass().getName().equals("[Ljava.lang.Integer;"))

{

int i;

x = obj[0];

for(i=0;i<n;i++) {

if((Integer)obj[i]>(Integer)x)

x=obj[i];

}

}

else

{

int i;

x = obj[0];

for(i=0;i<n;i++) {

if((Double)obj[i]>(Double)x)

x=obj[i];

}

}

return x;

}

}

public class MaxFind

{

public static void main(String[] args)

{

int choice,n;

TypeCheck<String> s1= null;

TypeCheck<Integer> s2= null;

TypeCheck<Double> s3= null;

String s[]= new String[100];

Integer in[]= new Integer[100];

Double d[] = new Double[100];

Scanner get = new Scanner(System.in);

System.out.print("Enter the size: ");

n=get.nextInt();

int i;

do {

System.out.print("Menu: 1.String 2.Integer 3.Double 4.Exit\nEnter the choice: ");

choice=get.nextInt();

switch(choice)

{

case 1:

for(i=0;i<n;i++)

{

System.out.print("Enter the string : " );

s[i]=get.next();

}

s1= new TypeCheck<String>(s,n);

System.out.println("Maximum value is :" + s1.Maximum());

break;

case 2:

for(i=0;i<n;i++)

{

System.out.print("Enter the Integer : " );

in[i]=get.nextInt();

}

s2= new TypeCheck<Integer>(in,n);

System.out.println("Maximum value is :" + s2.Maximum());

break;

case 3:

for(i=0;i<n;i++)

{

System.out.print("Enter the Double : " );

d[i]=get.nextDouble();

}

s3= new TypeCheck<Double>(d,n);

System.out.println("Maximum value is :" + s3.Maximum());

break;

}

}while(choice!=4);

}

}

**OUTPUT:**

Enter the size: 3

Menu: 1.String 2.Integer 3.Double 4.Exit

Enter the choice: 1

Enter the string : xya

Enter the string : xyz

Enter the string : abc

Maximum value is :xyz

Menu: 1.String 2.Integer 3.Double 4.Exit

Enter the choice: 2

Enter the Integer : 5

Enter the Integer : 6

Enter the Integer : 7

Maximum value is :7

Menu: 1.String 2.Integer 3.Double 4.Exit

Enter the choice: 3

Enter the Double : 4.5

Enter the Double : 5.5

Enter the Double : 1.5

Maximum value is :5.5

Menu: 1.String 2.Integer 3.Double 4.Exit

Enter the choice: 4

*//2. Write a Java program to create a generic stack and perform the operations.*

import java.util.Scanner;

class StackEmpty extends Exception

{

public String toString()

{

return "Stack is empty!!!";

}

}

class StackFull extends Exception

{

public String toString()

{

return "Stack is full!!!";

}

}

class Stack<t>

{

t arr[]=(t[])new Object[100];

int top;

int size;

Stack(int n)

{

top=-1;

int i;

for(i=0;i<n;i++)

arr[i]=(t)new Object();

size=n;

}

void push(t x) throws StackFull

{

if(top==size-1)

throw new StackFull();

else

arr[++top]=x;

}

t pop() throws StackEmpty

{

if(top==-1)

throw new StackEmpty();

else

{

return arr[top--];

}

}

void display()

{

int i;

for(i=top;i>=0;i--)

{

if(i==top)

System.out.println(arr[i]+" <---top");

else

System.out.println(arr[i]);

}

}

}

public class StackImpl

{

public static void main(String[] args)

{

int ch;

Integer a;Double b;String c;

Scanner get = new Scanner(System.in);

do

{

System.out.print("\nSelect the stack data type: 1.Integer 2.Double 3.String 4.exit \nenter your choice:");

ch=get.nextInt();

System.out.print("Enter the size: ");

int n=get.nextInt();

Stack<Integer> s1=null;

Stack<Double> s2=null;

Stack<String> s3=null;

switch(ch)

{

case 1:

s1 = new Stack<Integer>(n);

break;

case 2:

s2 = new Stack<Double>(n);

case 3:

s3= new Stack<String>(n);

}

if(ch==1)

{

int ch1 = 0;

do {

try {

System.out.print("\nOptions : 1.Push 2.Pop 3.Display 4.Exit\nEnter choice:");

ch1=get.nextInt();

switch(ch1) {

case 1:

System.out.print("Enter the element to be Inserted: ");

a=get.nextInt();

s1.push(a);

break;

case 2:

a=s1.pop();

System.out.println("Popped element is : "+a);

break;

case 3:

System.out.println("The elements of the Stack are: ");

s1.display();

}

}

catch(StackFull e)

{

System.out.println(e);

}

catch(StackEmpty e)

{

System.out.println(e);

}

}while(ch1!=4);

}

if(ch==2)

{

int ch2 = 0;

do {

try {

System.out.print("\nOptions : 1.Push 2.Pop 3.Display 4.Exit\nEnter choice:");

ch2=get.nextInt();

switch(ch2) {

case 1:

System.out.print("Enter the element to be Inserted: ");

b=get.nextDouble();

s2.push(b);

break;

case 2:

b=s2.pop();

System.out.println("Popped element is : "+b);

break;

case 3:

System.out.println("The elements of the Stack are: ");

s2.display();

}

}

catch(StackFull e)

{

System.out.println(e);

}

catch(StackEmpty e)

{

System.out.println(e);

}

}while(ch2!=4);

}

if(ch==3)

{

int ch3 = 0;

do {

try {

System.out.print("\nOptions : 1.Push 2.Pop 3.Display 4.Exit\nEnter choice:");

ch3=get.nextInt();

switch(ch3) {

case 1:

System.out.print("Enter the element to be Inserted: ");

c=get.next();

s3.push(c);

break;

case 2:

c=s3.pop();

System.out.println("Popped element is : "+c);

break;

case 3:

System.out.println("The elements of the Stack are: ");

s3.display();

}

}

catch(StackFull e)

{

System.out.println(e);

}

catch(StackEmpty e)

{

System.out.println(e);

}

}while(ch3!=4);

}

}while(ch!=4);

}

}

**OUTPUT:**

Select the stack data type: 1.Integer 2.Double 3.String 4.exit

enter your choice:1

Enter the size: 3

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: 1

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: 2

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: 3

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: 4

Stack is full!!!

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:3

The elements of the Stack are:

3 <---top

2

1

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Popped element is : 3

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Popped element is : 2

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Popped element is : 1

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Stack is empty!!!

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:4

Select the stack data type: 1.Integer 2.Double 3.String 4.exit

enter your choice:2

Enter the size: 3

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: 1.5

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: 2.5

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: 3.5

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: 4.5

Stack is full!!!

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:3

The elements of the Stack are:

3.5 <---top

2.5

1.5

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Popped element is : 3.5

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Popped element is : 2.5

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Popped element is : 1.5

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Stack is empty!!!

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:4

Select the stack data type: 1.Integer 2.Double 3.String 4.exit

enter your choice:3

Enter the size: 3

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: a

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: b

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: c

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:1

Enter the element to be Inserted: d

Stack is full!!!

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:3

The elements of the Stack are:

c <---top

b

a

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Popped element is : c

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Popped element is : b

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:2

Popped element is : a

Options : 1.Push 2.Pop 3.Display 4.Exit

Enter choice:4

Select the stack data type: 1.Integer 2.Double 3.String 4.exit

enter your choice:4

*//3. Write a Java program to perform a sorting operation on various types of elements*

*//using a generic method.*

*import java.util.Scanner;*

*class TypeCheck<T>*

*{ T obj[];*

*int n;*

*TypeCheck(T obj[], int n)*

*{*

*this.obj=obj;*

*this.n=n;*

*}*

*void sort()*

*{*

*T x;*

*if(obj.getClass().getName().equals("[Ljava.lang.String;"))*

*{*

*int i,j;*

*x=obj[0];*

*for(i=0;i<n-1;i++)*

*{ for(j=0;j<n-i-1;j++)*

*if(((String) obj[j]).compareTo((String) obj[j+1])>0)*

*{ x=obj[j];*

*obj[j]=obj[j+1];*

*obj[j+1]=x;*

*}*

*}*

*System.out.println("sorted array:");*

*for(i=0;i<n;i++)*

*System.out.println(obj[i]);*

*}*

*else if(obj.getClass().getName().equals("[Ljava.lang.Integer;"))*

*{*

*int i,j;*

*x=obj[0];*

*for(i=0;i<n-1;i++)*

*{ for(j=0;j<n-i-1;j++)*

*if(((Integer) obj[j]).compareTo((Integer) obj[j+1])>0)*

*{ x=obj[j];*

*obj[j]=obj[j+1];*

*obj[j+1]=x;*

*}*

*}*

*System.out.println("sorted array:");*

*for(i=0;i<n;i++)*

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

*}*

*else*

*{*

*int i,j;*

*x=obj[0];*

*for(i=0;i<n-1;i++)*

*{ for(j=0;j<n-i-1;j++)*

*if(((Double) obj[j]).compareTo((Double) obj[j+1])>0)*

*{ x=obj[j];*

*obj[j]=obj[j+1];*

*obj[j+1]=x;*

*}*

*}*

*System.out.println("sorted array:");*

*for(i=0;i<n;i++)*

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

*}*

*}*

*}*

*public class Sorted*

*{*

*public static void main(String[] args)*

*{*

*int choice,n;*

*TypeCheck<String> s1= null;*

*TypeCheck<Integer> s2= null;*

*TypeCheck<Double> s3= null;*

*String s[]= new String[100];*

*Integer in[]= new Integer[100];*

*Double d[] = new Double[100];*

*Scanner get = new Scanner(System.in);*

*System.out.print("Enter the size: ");*

*n=get.nextInt();*

*int i;*

*do {*

*System.out.print("Menu: 1.String 2.Integer 3.Double 4.Exit\nEnter the choice: ");*

*choice=get.nextInt();*

*switch(choice)*

*{*

*case 1:*

*for(i=0;i<n;i++)*

*{*

*System.out.print("Enter the string : " );*

*s[i]=get.next();*

*}*

*s1= new TypeCheck<String>(s,n);*

*s1.sort();*

*break;*

*case 2:*

*for(i=0;i<n;i++)*

*{*

*System.out.print("Enter the Integer : " );*

*in[i]=get.nextInt();*

*}*

*s2= new TypeCheck<Integer>(in,n);*

*s2.sort();*

*break;*

*case 3:*

*for(i=0;i<n;i++)*

*{*

*System.out.print("Enter the Double : " );*

*d[i]=get.nextDouble();*

*}*

*s3= new TypeCheck<Double>(d,n);*

*s3.sort();*

*break;*

*}*

*}while(choice!=4);*

*}*

*}*

***OUTPUT:***

*Enter the size: 3*

*Menu: 1.String 2.Integer 3.Double 4.Exit*

*Enter the choice: 1*

*Enter the string : abc*

*Enter the string : yo*

*Enter the string : bye*

*sorted array:*

*abc*

*bye*

*yo*

*Menu: 1.String 2.Integer 3.Double 4.Exit*

*Enter the choice: 2*

*Enter the Integer : 3*

*Enter the Integer : 7*

*Enter the Integer : 5*

*sorted array:*

*3 5 7*

*Menu: 1.String 2.Integer 3.Double 4.Exit*

*Enter the choice: 3*

*Enter the Double : 5.5*

*Enter the Double : 3.5*

*Enter the Double : 2.5*

*sorted array:*

*2.5 3.5 5.5*

*Menu: 1.String 2.Integer 3.Double 4.Exit*

*Enter the choice: 4*