**Ex7 - Generics**

**-------------------------------------------------------------------------------------------------------------------------------**

Name: Aviansh Gupta

Reg No.: 185001028

CSE - A

**-------------------------------------------------------------------------------------------------------------------------------**

**1. Write a java program to create a generic stack and perform the operations.**

import java.util.\*;

class GenericStack<T>{

int topofstack=0,maxsize;

final int emptytos = 0;

T []stack;

public GenericStack(int maxsize,T []stack){

this.maxsize = maxsize;

this.stack = stack;

}

public boolean isFull(){

return topofstack == maxsize;

}

public boolean isEmpty(){

return topofstack == emptytos;

}

public void push(T element){

if(isFull())

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

else

stack[++topofstack] = element;

}

public T pop(){

if(isEmpty()){

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

return null;

}

return stack[topofstack--];

}

public T peek(){

if(isEmpty()){

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

return null;

}

return stack[topofstack];

}

void display(){

int ind = 1;

while(ind <= topofstack)

System.out.print("---->"+stack[ind++]);

System.out.println();

}

}

class TestGenericStack{

public static void main(String []args){

Scanner sc = new Scanner(System.in);

//System.out.println("Enter the number of elements in the stack( < 15): ");

//int num = sc.nextInt();

int ch1,ch2;

int maxsize = 20;

System.out.println("DATA TYPES\n1: String\n2: Character\n3: Number");

System.out.print("\nEnter your choice: ");

ch1 = sc.nextInt();

do{

switch(ch1){

case 1: String []arr = new String[maxsize];

GenericStack<String> gs1 = new GenericStack<String>(maxsize,arr);

System.out.println("STACK OPERATIONS\n1: Push\n2: Pop\n3: Display\n4: Peek");

System.out.print("\nEnter your choice: ");

ch2 = sc.nextInt();

do{

sc.nextLine();

switch(ch2){

case 1: System.out.print("Enter the element to be pushed:");

gs1.push(sc.nextLine());

break;

case 2: System.out.println("Popped element: "+gs1.pop());

break;

case 3: gs1.display();

break;

case 4: System.out.println("Top element: "+gs1.peek());

break;

default: System.out.println("Enter a valid choice");

break;

}

System.out.println("STACK OPERATIONS\n1: Push\n2: Pop\n3: Display\n4: Peek");

System.out.print("\nEnter your choice(To stop enter -1): ");

ch2 = sc.nextInt();

}while(ch2 != -1);

break;

case 2: Character []arr1 = new Character[maxsize];

GenericStack<Character> gs2 = new GenericStack<Character>(maxsize,arr1);

System.out.println("STACK OPERATIONS\n1: Push\n2: Pop\n3: Display\n4: Peek");

System.out.print("\nEnter your choice: ");

ch2 = sc.nextInt();

do{

switch(ch2){

case 1: System.out.print("Enter the element to be pushed:");

gs2.push(sc.next().charAt(0));

break;

case 2: System.out.println("Popped element: "+gs2.pop());

break;

case 3: gs2.display();

break;

case 4: System.out.println("Top element: "+gs2.peek());

break;

default: System.out.println("Enter a valid choice");

break;

}

System.out.println("STACK OPERATIONS\n1: Push\n2: Pop\n3: Display\n4: Peek");

System.out.print("\nEnter your choice(To stop enter -1): ");

ch2 = sc.nextInt();

}while(ch2 != -1);

break;

case 3: Number []arr2 = new Number[maxsize];

GenericStack<Number> gs3 = new GenericStack<Number>(maxsize,arr2);

System.out.println("STACK OPERATIONS\n1: Push\n2: Pop\n3: Display\n4: Peek");

System.out.print("\nEnter your choice: ");

ch2 = sc.nextInt();

do{

switch(ch2){

case 1: System.out.print("Enter the element to be pushed:");

gs3.push(sc.nextDouble());

break;

case 2: System.out.printf("Popped element: %.3f\n",gs3.pop());

break;

case 3: gs3.display();

break;

case 4: System.out.printf("Top element: %.3f\n",gs3.peek());

break;

default: System.out.println("Enter a valid choice");

continue;

}

System.out.println("STACK OPERATIONS\n1: Push\n2: Pop\n3: Display\n4: Peek");

System.out.print("\nEnter your choice(To stop enter -1): ");

ch2 = sc.nextInt();

}while(ch2 != -1);

break;

default: System.out.println("Enter a valid choice");

continue;

}

System.out.println("DATA TYPES\n1: String\n2: Character\n3: Number");

System.out.print("\nEnter your choice(To stop enter -1): ");

ch1 = sc.nextInt();

}while(ch1 != -1);

}

}

**----------------------------------------------------------------------**

**OUTPUT**

DATA TYPES

1: String

2: Character

3: Number

Enter your choice: 1

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice: 1

Enter the element to be pushed:Marques

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 1

Enter the element to be pushed:John

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 1

Enter the element to be pushed:Peter

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 1

Enter the element to be pushed:Tyler

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 3

---->Marques---->John---->Peter---->Tyler

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 2

Popped element: Tyler

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 4

Top element: Peter

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 3

---->Marques---->John---->Peter

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): -1

DATA TYPES

1: String

2: Character

3: Number

Enter your choice(To stop enter -1): 2

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice: 1

Enter the element to be pushed:a

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 1

Enter the element to be pushed:b

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 1

Enter the element to be pushed:c

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 1

Enter the element to be pushed:d

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 3

---->a---->b---->c---->d

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 2

Popped element: d

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 4

Top element: c

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 3

---->a---->b---->c

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): -1

DATA TYPES

1: String

2: Character

3: Number

Enter your choice(To stop enter -1): 3

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice: 1

Enter the element to be pushed:28

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 1

Enter the element to be pushed:31

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 1

Enter the element to be pushed:40

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 1

Enter the element to be pushed:7

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 3

---->28.0---->31.0---->40.0---->7.0

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 2

Popped element: 7.000

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 4

Top element: 40.000

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): 3

---->28.0---->31.0---->40.0

STACK OPERATIONS

1: Push

2: Pop

3: Display

4: Peek

Enter your choice(To stop enter -1): -1

DATA TYPES

1: String

2: Character

3: Number

Enter your choice(To stop enter -1): -1

**-------------------------------------------------------------------------------------------------------------------------------**

**2. Write a java program to find the maximum value from the given type of elements using a generic function.**

import java.util.\*;

class GenericMax{

<T extends Comparable> T findMax(T []arr,int []out){

T max = arr[0];

int i;

for(i = 1;i < arr.length;i++){

if(max.compareTo(arr[i])<0){

out[0] = i;

max = arr[i];

}

}

return max;

}

}

class TestGenaricMax{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

GenericMax gm = new GenericMax();

int choice,num;

int []out = new int[1];

System.out.println("DATA TYPES\n1: String\n2: Integer\n3: Float");

System.out.print("Enter Your Choice: ");

choice = sc.nextInt();

do{

switch(choice){

case 1: System.out.print("Enter the number of elements: ");

num = sc.nextInt();

String []arr = new String[num];

System.out.println("Enter the elements: ");

sc.nextLine();

for(int i=0;i<num;i++)

arr[i] = sc.nextLine();

String max = gm.<String>findMax(arr,out);

System.out.printf("\nMax element: %s\tMax Index: %d\n\n",max,out[0]);

break;

case 2: System.out.print("Enter the number of elements: ");

num = sc.nextInt();

Integer []arr1 = new Integer[num];

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

sc.nextLine();

for(int i=0;i<num;i++)

arr1[i] = sc.nextInt();

Integer max1 = gm.<Integer>findMax(arr1,out);

System.out.printf("\nMax element: %d\tMax Index: %d\n\n",max1,out[0]);

break;

case 3: System.out.print("Enter the number of elements: ");

num = sc.nextInt();

Float []arr2 = new Float[num];

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

sc.nextLine();

for(int i=0;i<num;i++)

arr2[i] = sc.nextFloat();

Float max2 = gm.<Float>findMax(arr2,out);

System.out.printf("\nMax element: %.3f\tMax Index: %d\n\n",max2,out[0]);

break;

}

System.out.println("DATA TYPES\n1: String\n2: Integer\n3: Float");

System.out.print("Enter Your Choice(To stop enter -1): ");

choice = sc.nextInt();

}while(choice != -1);

}

}

**----------------------------------------------------------------------**

**OUTPUT**

DATA TYPES

1: String

2: Integer

3: Float

Enter Your Choice: 1

Enter the number of elements: 5

Enter the elements:

John

Peter

Dave

Marques

Tyler

Max element: Tyler Max Index: 4

DATA TYPES

1: String

2: Integer

3: Float

Enter Your Choice(To stop enter -1): 2

Enter the number of elements: 5

Enter the elements: 20 75 153 22 98

Max element: 153 Max Index: 2

DATA TYPES

1: String

2: Integer

3: Float

Enter Your Choice(To stop enter -1): 3

Enter the number of elements: 5

Enter the elements: 2.22 2.21 2.98 3.01 0.12

Max element: 3.01 Max Index: 3

DATA TYPES

1: String

2: Integer

3: Float

Enter Your Choice(To stop enter -1): -1

**-------------------------------------------------------------------------------------------------------------------------------3. Perform a sorting operation on various types of elements using generic method.**

import java.util.\*;

class GenericSort{

<T extends Comparable> void isort(T []arr){

for(int pos=1;pos<arr.length;pos++){

int nextpos = pos;

while(nextpos > 0 && arr[nextpos].compareTo(arr[nextpos-1]) < 0){

T temp = arr[nextpos];

arr[nextpos] = arr[nextpos - 1];

arr[nextpos - 1] = temp;

nextpos--;

}

}

}

}

class TestGenericSort{

public static void main(String []args){

Scanner sc = new Scanner(System.in);

GenericSort gs = new GenericSort();

int choice,num;

System.out.println("DATA TYPES\n1: String\n2: Integer\n3: Float");

System.out.print("Enter Your Choice: ");

choice = sc.nextInt();

do{

switch(choice){

case 1: System.out.print("Enter the number of elements: ");

num = sc.nextInt();

String []arr = new String[num];

System.out.println("Enter the elements: ");

sc.nextLine();

for(int i=0;i<num;i++)

arr[i] = sc.nextLine();

gs.<String>isort(arr);

System.out.print("\nSorted Array: ");

for(int i=0;i<num;i++)

System.out.printf("%s ",arr[i]);

System.out.println("\n");

break;

case 2: System.out.print("Enter the number of elements: ");

num = sc.nextInt();

Integer []arr1 = new Integer[num];

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

sc.nextLine();

for(int i=0;i<num;i++)

arr1[i] = sc.nextInt();

gs.<Integer>isort(arr1);

System.out.print("\nSorted Array: ");

for(int i=0;i<num;i++)

System.out.printf("%d ",arr1[i]);

System.out.println("\n");

break;

case 3: System.out.print("Enter the number of elements: ");

num = sc.nextInt();

Float []arr2 = new Float[num];

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

sc.nextLine();

for(int i=0;i<num;i++)

arr2[i] = sc.nextFloat();

gs.<Float>isort(arr2);

System.out.print("\nSorted Array: ");

for(int i=0;i<num;i++)

System.out.printf("%.2f ",arr2[i]);

System.out.println("\n");

break;

}

System.out.println("DATA TYPES\n1: String\n2: Integer\n3: Float");

System.out.print("Enter Your Choice(To stop enter -1): ");

choice = sc.nextInt();

}while(choice != -1);

}

}

**----------------------------------------------------------------------OUTPUT**

DATA TYPES

1: String

2: Integer

3: Float

Enter Your Choice: 2

Enter the number of elements: 7

Enter the elements: 28 10 33 25 67 22 99

Sorted Array: 10 22 25 28 33 67 99

DATA TYPES

1: String

2: Integer

3: Float

Enter Your Choice(To stop enter -1): 3

Enter the number of elements: 5

Enter the elements: 2.22 2.21 2.29 3.11 0.29

Sorted Array: 0.29 2.21 2.22 2.29 3.11

DATA TYPES

1: String

2: Integer

3: Float

Enter Your Choice(To stop enter -1): 1

Enter the number of elements: 5

Enter the elements:

John

Marques

Dave

Peter

Tyler

Sorted Array: Dave John Marques Peter Tyler

DATA TYPES

1: String

2: Integer

3: Float

Enter Your Choice(To stop enter -1): -1