Assignment – 6: Generic Types

=======================================================================

Objective:

To implement generic types – generic classes and methods

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

generic function.

import java.util.Scanner;

public class GenFunc{

public static <T extends Number> T maxOf(T[] arr){

T max=arr[0];

for(int i=0;i<arr.length;i++)

if(max.doubleValue()<arr[i].doubleValue())

max=arr[i];

return max;

}

public static void main(String args[]){

int option,n;

Scanner in=new Scanner(System.in);

do{

System.out.println("Choose data type 1-Integer 2-Double ");

System.out.print(" Your choice: ");option=in.nextInt();

if(option==1){

System.out.print("Enter size: ");n=in.nextInt();

Number arr[]=new Integer[n];

System.out.println("Enter the array ");

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

arr[i]=in.nextInt();

System.out.println("\n The maximum value is "+GenFunc.maxOf(arr));

}

else if(option==2){

System.out.print("Enter size: ");n=in.nextInt();

Number arr[]=new Double[n];

System.out.println("Enter the array ");

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

arr[i]=in.nextDouble();

System.out.println("\n The maximum value is "+GenFunc.maxOf(arr));

}

else if(option !=0)

System.out.println("Invalid option");

else;

}while(option!=0);

}

}

/\*

Output:

Choose data type 1-Integer 2-Double

Your choice: 1

Enter size: 5

Enter the array

1 2 3 4 5

The maximum value is 5

Choose data type 1-Integer 2-Double

Your choice: 2

Enter size: 5

Enter the array

5 7 8 9 6 1

The maximum value is 9.0

Choose data type 1-Integer 2-Double

Your choice: Enter size: 0

\*/

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

import java.util.Scanner;

public class GenSort{

public static <T extends Number> void sort(T[] arr){

for(int i=0;i<arr.length;i++){

int min=i;

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

if(arr[min].doubleValue()>arr[j].doubleValue())

min=j;

T tmp=arr[i];

arr[i]=arr[min];

arr[min]=tmp;

}

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

for(int i=0;i<arr.length;i++)

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

}

public static void main(String args[]){

int option,n;

Scanner in=new Scanner(System.in);

do{

System.out.println("\nChoose data type 1-Integer 2-Double ");

System.out.print(" Your choice: ");option=in.nextInt();

if(option==1){

System.out.print("Enter size: ");n=in.nextInt();

Number arr[]=new Integer[n];

System.out.println("Enter the array ");

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

arr[i]=in.nextInt();

System.out.print("\nUnsorted array: ");

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

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

GenSort.sort(arr);

}

else if(option==2){

System.out.print("Enter size: ");n=in.nextInt();

Number arr[]=new Double[n];

System.out.println("Enter the array ");

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

arr[i]=in.nextDouble();

System.out.print("\nUnsorted array: ");

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

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

GenSort.sort(arr);

}

else if(option !=0)

System.out.println("Invalid option");

else;

}while(option!=0);

}

}

/\*

Output:

Choose data type 1-Integer 2-Double

Your choice: 1

Enter size: 5

Enter the array

6 8 4 9 2

Unsorted array: 6 8 4 9 2

Sorted array: 2 4 6 8 9

Choose data type 1-Integer 2-Double

Your choice: 2

Enter size: 5

Enter the array

6 8 4 9 2

Unsorted array: 6.0 8.0 4.0 9.0 2.0

Sorted array: 2.0 4.0 6.0 8.0 9.0

Choose data type 1-Integer 2-Double

Your choice: 0

\*/

3. Write a Java program to perform a sorting operation on various types of elements using a generic

method.

import java.util.Scanner;

class StackFullException extends Exception{

public String toString(){

return "Stack is Full";

}

}

class StackEmptyException extends Exception{

public String toString(){

return "Stack is Empty";

}

}

class StackImpl<T>{

T[] data;

int top,limit;

StackImpl(int limit,T[] arr){

data=arr;

top=-1;

this.limit=limit;

}

void pushStack(T x) throws StackFullException{

if(top==limit-1)

throw new StackFullException();

else{

data[++top]=x;

System.out.print("\n Push Attempt Successful \n");

}

}

void popStack() throws StackEmptyException{

if(top==-1)

throw new StackEmptyException();

else{

System.out.println("\n"+data[top--]+" has been popped ");

System.out.print("\n Pop Attempt Successful \n");

}

}

void displayElements(){

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

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

}

}

public class GenStack{

public static void main(String args[]){

int limit;

Scanner in=new Scanner(System.in);

System.out.print(" Enter the capacity of the stack ");

limit=in.nextInt();

int dataType;

do{

System.out.print("\n Choose data type\n 1-Integer\n 2-Double\n Your choice: ");dataType=in.nextInt();

}while(dataType!=1 && dataType!=2);

Integer iarr[]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

Double darr[]={0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};

StackImpl<Integer> iS=new StackImpl<Integer>(limit,iarr);

StackImpl<Double> dS=new StackImpl<Double>(limit,darr);

int option;

do{

System.out.print("\n Wassup?\n 1-push\n 2-pop\n 3-display\n 0-exit\n");

System.out.print(" Your choice: ");option=in.nextInt();

if(option==1){

if(dataType==1){

int x;

System.out.print("\n Enter element to push: ");x=in.nextInt();

try{

iS.pushStack(x);

}

catch(StackFullException sfe){

System.out.print("\n"+sfe.toString()+". Cannot perform push operation \n");

}

}

else if(dataType==2){

double x;

System.out.print("\n Enter element to push: ");x=in.nextDouble();

try{

dS.pushStack(x);

}

catch(StackFullException sfe){

System.out.print("\n"+sfe.toString()+". Cannot perform push operation \n");

}

}

}

else if(option==2){

try{

if(dataType==1)

iS.popStack();

else

dS.popStack();

}

catch(StackEmptyException see){

System.out.print("\n"+see.toString()+". Cannot perform pop operation \n");

}

}

else if(option==3){

if(dataType==1)

iS.displayElements();

else

dS.displayElements();

}

else if(option!=0)

System.out.println("\nInvalid option ");

else;

}while(option!=0);

}

}

/\*

Output:

Enter the capacity of the stack 5

Choose data type

1-Integer

2-Double

Your choice: 1

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 1

Enter element to push: 1

Push Attempt Successful

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 1

Enter element to push: 2

Push Attempt Successful

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 1

Enter element to push: 3

Push Attempt Successful

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 1

Enter element to push: 4

Push Attempt Successful

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 1

Enter element to push: 5

Push Attempt Successful

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 1

Enter element to push: 6

Stack is Full. Cannot perform push operation

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 3

1 2 3 4 5

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 2

5 has been popped

Pop Attempt Successful

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 2

4 has been popped

Pop Attempt Successful

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 2

3 has been popped

Pop Attempt Successful

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 2

2 has been popped

Pop Attempt Successful

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 2

1 has been popped

Pop Attempt Successful

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 2

Stack is Empty. Cannot perform pop operation

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 3

Wassup?

1-push

2-pop

3-display

0-exit

Your choice: 0

\*/