Greg Witt

C202

GenericStack Object

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

package genericclass;

public class GenericStack<E> {

public java.util.ArrayList<E> list = new java.util.ArrayList<>();

public void push(E O){

list.add(O);

}//push

public int getSize(){

return list.size();

}//getSize

public E peek(){

return list.get(getSize() -1);

}//peek

public <E> boolean isEmpty(){

return list.isEmpty();

}//isEmpty

public E pop(){

E o = list.get(getSize() - 1);

list.remove(getSize() - 1);

return o;

}//pop

@Override

public String toString(){

return "Stack: " + list.toString();

}//toString

public <E extends Comparable<E>> E Max(E o1, E o2){

if(o1.compareTo(o2) > 0)

return o1;

else

return o2;

}//max

}//class

==================================================== GenericRectangle Object

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

package genericclass;

public class GenericRectangle<T extends Number> {

T height;

T width;

public GenericRectangle(T h, T w){

height = h;

width = w;

}//DefaultConstructor

public Number getArea(){

return (height.doubleValue() \* width.doubleValue());

}//getArea

}//GenericRectangle

GenericClients

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

package genericclass;

public class GenericClass {

public static void main(String[] args) {

GenericStack<Integer> ans = new GenericStack<>();

ans.push(4);

ans.push(8);

System.out.println(ans.getSize());

System.out.println(ans.list);

System.out.println(ans.peek());

GenericRectangle<Integer> r1 = new GenericRectangle<>(8,2);

System.out.println(r1.getArea());

}//main

}//class

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

Runtime Output

run:

2

[4, 8]

8

16.0

BUILD SUCCESSFUL (total time: 0 seconds)