CS2002-1

Lab Programs by
Deekshi R Gowda
NNM24IS064

Submitted to: Dr. Martis

Lab Program 5:

Case:

A local ticket counter maintains a stack of Persons who are standing in line. Each Person has:

- name (String)
- age (int)
- gender (enum: MALE, FEMALE, OTHER)

The counter wants you to implement a Stack of Persons with the following requirements:

1. Person Class

- Attributes: name, age, gender
- Constructor(s) to initialize Person
- Method: displayPerson() → prints details of that person

2. Stack Class

- A stack that stores Person objects (not just integers).
- Two constructors:

o Stack(int size) → creates empty stack of given size

o Stack(Person[] arr) → initializes stack directly from array of Persons

- Push Operation (Overloaded):

o push(Person p) → Push single Person

o push(Person p1, Person p2) → Push two Persons together

- Pop Operation (Overloaded):

o pop() → Pop one Person

o pop(int n) → Pop n Persons

- Display Operation (Overloaded):

o display() → Show full stack (all Persons with details)

o display(int n) → Show only top n Persons

3. Main Program (Menu Driven)

Allow user to:

1. Push one Person (enter details: name, age, gender)

2. Push two Persons (enter both details)

CS2002-1 (Object Oriented Programming) 2 | Page

3. Pop one Person

4. Pop multiple Persons

5. Display all Persons in stack

6. Display top n Persons

7. Exit

GitHub Link: https://github.com/mgkrupa06/Java_Progs/tree/main/Prog5

CODE:

```
package stackperson;
enum Gender {
MALE, FEMALE, OTHER;
}
package stackperson;
class Person {
private String name;
private int age;
private Gender gender;
Person(String name, int age, Gender gender) {
this.name = name;
this.age = age;
this.gender = gender;
}
```

```
void displayPerson() {
System.out.println("Name: " + name);
System.out.println("Age: " + age);
System.out.println("Gender: " + gender);
}
}
package stackperson;
class Stack {
private Person arr[];
private int top;
private int capacity;
Stack(int size) {
capacity = size;
arr = new Person[capacity];
top = -1;
}
Stack(Person[] inputArr) {
capacity = inputArr.length;
arr = new Person[capacity];
for (int i = 0; i < capacity; i++) {
arr[i] = inputArr[i];
}
top = capacity - 1;
}
void push(Person p) {
if (top >= capacity - 1) {
System.out.println("Stack Overflow");
return;
}
arr[++top] = p;
```

```
System.out.println("Person pushed successfully");
}
void push(Person p1, Person p2) {
push(p1);
push(p2);
}
Person pop() {
if (top == -1) {
System.out.println("Stack Underflow");
return null;
}
return arr[top--];
}
void pop(int n) {
if (n <= 0) {
System.out.println("Invalid number of Persons to pop");
return;
}
for (int i = 0; i < n; i++) {
Person p = pop();
if (p == null) break;
System.out.println("Popped Person:");
p.displayPerson();
}
}
void display() {
if (top == -1) {
System.out.println("Stack is empty");
return;
}
```

```
System.out.println("Persons in stack (Top to Bottom):");
for (int i = top; i >= 0; i--) {
arr[i].displayPerson();
}
}
void display(int n) {
if (top == -1) {
System.out.println("Stack is empty");
return;
}
if (n <= 0) {
System.out.println("Invalid number of Persons to display");
return;
}
System.out.println("Top " + n + " Persons:");
for (int i = top; i >= 0 && i > top - n; i--) {
arr[i].displayPerson();
}
}
}
package stackperson;
import java.util.*;
public class TicketCounterStack {
public static Person readPerson(Scanner sc) {
sc.nextLine();
System.out.print("Enter name: ");
String name = sc.nextLine();
System.out.print("Enter age: ");
int age = sc.nextInt();
System.out.print("Enter gender (MALE, FEMALE, OTHER): ");
```

```
sc.nextLine();
String genderStr = sc.nextLine().toUpperCase();
Gender gender;
try {
gender = Gender.valueOf(genderStr);
} catch (IllegalArgumentException e) {
System.out.println("Invalid gender");
gender = Gender.OTHER;
}
return new Person(name, age, gender);
}
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
Stack st = null;
System.out.println("Choose initialization:");
System.out.println("1. Empty stack with size");
System.out.println("2. Stack initialized with array of Persons");
int choice = sc.nextInt();
if (choice == 1) {
System.out.print("Enter size: ");
int size = sc.nextInt();
st = new Stack(size);
} else if (choice == 2) {
System.out.print("Enter number of Persons: ");
int n = sc.nextInt();
Person arr[] = new Person[n];
for (int i = 0; i < n; i++) {
System.out.println("Enter details for Person " + (i + 1) + ":");
arr[i] = readPerson(sc);
}
```

```
st = new Stack(arr);
} else {
System.out.println("Invalid choice");
return;
}
int option;
do {
System.out.println("\n Menu ");
System.out.println("1. Push one Person");
System.out.println("2. Push two Persons");
System.out.println("3. Pop one Person");
System.out.println("4. Pop multiple Persons");
System.out.println("5. Display all Persons");
System.out.println("6. Display top n Persons");
System.out.println("7. Exit");
System.out.print("Enter your choice: ");
option = sc.nextInt();
switch(option) {
case 1:
System.out.println("Enter details for Person:");
Person p1 = readPerson(sc);
st.push(p1);
break;
case 2:
System.out.println("Enter details for first Person:");
Person p2a = readPerson(sc);
System.out.println("Enter details for second Person:");
Person p2b = readPerson(sc);
st.push(p2a, p2b);
break;
case 3: Person popped = st.pop();
if (popped != null) {
System.out.println("Popped Person:");
popped.displayPerson();
}
break;
case 4:
System.out.print("Enter number of Persons to pop: ");
int n = sc.nextInt();
st.pop(n);
break;
case 5:
st.display();
break;
case 6:
System.out.print("Enter number of top Persons: ");
int topN = sc.nextInt();
st.display(topN);
break;
case 7:
System.out.println("Exiting...");
break;
default:
System.out.println("Invalid choice!");
CS2002-1 (Object Oriented Programming) 10 | P a g e
}
} while(option != 7);
}
}
```

OUTPUT:

```
Choose initialization:
1. Empty stack with size
2. Stack initialized with array of Persons
1
Enter size: 3
Menu
1. Push one Person
2. Push two Persons
3. Pop one Person
4. Pop multiple Persons
5. Display all Persons
6. Display top n Persons
7. Exit
Enter your choice: 1
Enter details for Person:
Enter name: Deekshi
Enter age: 19
Enter gender (MALE, FEMALE, OTHER): FEMALE
Person pushed successfully
Menu
1. Push one Person
2. Push two Persons
3. Pop one Person
4. Pop multiple Persons
5. Display all Persons
6. Display top n Persons
7. Exit
Enter your choice: 2
Enter details for first Person:
Enter name: Preethika
Enter age: 20
Enter gender (MALE, FEMALE, OTHER): FEMALE
Enter details for second Person:
Enter name: Suhas
Enter age: 21
Enter gender (MALE, FEMALE, OTHER): MALE
Person pushed successfully
Person pushed successfully
Menu
1. Push one Person
2. Push two Persons
3. Pop one Person
4. Pop multiple Persons
5. Display all Persons
6. Display top n Persons
7. Exit
Enter your choice: 5
Persons in stack (Top to Bottom):
Name: Suhas
Age: 21
Gender: MALE
Name: Preethika
Age: 20
Gender: FEMALE
Name: Deekshi
Age: 19
Gender: FEMALE
Menu
1. Push one Person
2. Push two Persons
```

```
3. Pop one Person
4. Pop multiple Persons
5. Display all Persons
6. Display top n Persons
7. Exit
Enter your choice: 6
Enter number of top Persons: 1
Top 1 Persons:
Name: Suhas
Age: 21
Gender: MALE
Menu
1. Push one Person
2. Push two Persons
3. Pop one Person
4. Pop multiple Persons
5. Display all Persons
6. Display top n Persons
7. Exit
Enter your choice: 4
Enter number of Persons to pop: 1
Popped Person:
Name: Suhas
Age: 21
Gender: MALE
Menu
1. Push one Person
2. Push two Persons
3. Pop one Person
4. Pop multiple Persons
```

```
5. Display all Persons
6. Display top n Persons
7. Exit
Enter your choice: 5
Persons in stack (Top to Bottom):
Name: Preethika
Age: 20
Gender: FEMALE
Name: Deekshi
Age: 19
Gender: FEMALE
Menu
1. Push one Person
2. Push two Persons
3. Pop one Person
4. Pop multiple Persons
5. Display all Persons
6. Display top n Persons
7. Exit
Enter your choice: 7
Exiting...
Choose initialization:
1. Empty stack with size
2. Stack initialized with array of Persons
2
Enter number of Persons: 3
Enter details for Person 1:
Enter name: Deekshi
Enter age: 19
Enter gender (MALE, FEMALE, OTHER): FEMALE
Enter details for Person 2:
Enter name: Preethika
Enter age: 20
Enter gender (MALE, FEMALE, OTHER): FEMALE
Enter details for Person 3:
Enter name: Suhas
Enter age: 21
Enter gender (MALE, FEMALE, OTHER): MALE
Menu
1. Push one Person
2. Push two Persons
3. Pop one Person
4. Pop multiple Persons
5. Display all Persons
6. Display top n Persons
7. Exit
Enter your choice: 1
Enter details for Person:
Enter name: Poorvi
Enter age: 21
Enter gender (MALE, FEMALE, OTHER): FEMALE
Stack Overflow
Menu
1. Push one Person
2. Push two Persons
3. Pop one Person
4. Pop multiple Persons
5. Display all Persons
6. Display top n Persons
7. Exit
```

```
Enter your choice: 3

Popped Person:

Name: Suhas

Age: 21

Gender: MALE

Menu

1. Push one Person

2. Push two Persons

3. Pop one Person

4. Pop multiple Persons

5. Display all Persons

6. Display top n Persons

7. Exit

Enter your choice: 5

Persons in stack (Top to Bottom):

Name: Preethika

Age: 20

Gender: FEMALE

Name: Deekshi

Age: 19

Gender: FEMALE

Menu

1. Push one Person

2. Push two Persons

3. Pop one Person

4. Pop multiple Persons

5. Display all Persons

6. Display top n Persons

7. Exit

Enter your choice: 6
Enter number of top Persons: 1

Top 1 Persons:

Name: Preethika

Age: 20

Gender: FEMALE

Menu

1. Push one Person

2. Push two Persons

3. Pop one Person

4. Pop multiple Persons

5. Display all Persons

6. Display top n Persons

7. Exit

Enter your choice: 3

Popped Person:

Name: Preethika

Age: 20

Gender: FEMALE

Menu

1. Push one Person

2. Push two Persons

3. Pop one Person

4. Pop multiple Persons

5. Display all Persons

6. Display top n Persons

7. Exit

Enter your choice: 5

Persons in stack (Top to Bottom):

Name: Deekshi

Age: 19
```

```
Gender: FEMALE

Menu

1. Push one Person

2. Push two Persons

3. Pop one Person

4. Pop multiple Persons

5. Display all Persons

6. Display top n Persons

7. Exit

Enter your choice: 7

Exiting...
```