JOBSHEET 8 QUEUE

Muhammad Tegar Hibatulloh

2341720221 / 20/ 1I

1.2.1. Steps

Queue.java: https://github.com/Garrss/Algoritma-Jobsheet/blob/main/Jobsheet10/Queue.java

QueueMain.java: https://github.com/Garrss/Algoritma-Jobsheet/blob/main/Jobsheet10/QueueMain.java

Result:

```
Insert maximum queue : 4
Choose menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=========================
1
Insert new data: 15
Choose menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=========================
1
Insert new data: 31
Choose menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=========================
4
The first element : 15
Choose menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=========================
6
```

1.2.3. Questions

1. In method create(), why is the front and rear attribute has initial value with 1 and not 0?
   - front and rear are initialized to -1 to signify an empty queue. -1 indicates an empty queue, not 1.
2. In method enqueue(), please explain the usage of this following code

```
if (rear == max - 1) {
    rear = 0;
```

   - if (rear == max - 1) checks if the rear pointer is at the end of the array. If so, it wraps around to the beginning by setting rear to 0.
3. Observe enqueue() method, which line of code indicates that the new data will be stored in last position of the queue?
   - New data is stored in the last position of the queue with Q[rear] = data;
4. Observe dequeue() method, which line of code indicates that the data is removed in the first position of the queue?
   - Data is removed from the first position of the queue with data = Q[front];
5. In dequeue method(), explain the usage of these codes !

```
if (front == max - 1) {
    front = 0;
```

   - If (front == max -1) handles wrapping the front pointer to the beginning of the array if it reaches the end.
6. In method print(), why the loop process has int i = 0 instead of int i=front?
   - The loop process in the print() method starts from int i = front; to ensure printing starts from the first element of the queue.
7. In method print(), please explain why we insert this code in our program?
   - The line i = (i + 1) % max; for correctly traversing the circular queue and printing its elements without going beyond the array bounds.

1.3.1. Steps

Passengers.java: https://github.com/Garrss/Algoritma-Jobsheet/blob/main/Jobsheet10/Passengers.java

QueuePassengers.java: https://github.com/Garrss/Algoritma-Jobsheet/blob/main/Jobsheet10/QueuePassengers.java

PassengersMain.java: https://github.com/Garrss/Algoritma-Jobsheet/blob/main/Jobsheet10/PassengersMain.java

Results:

```
Insert maximum queue : 5
Choose menu:
1. Queue
2. Dequeue
3. Check first queue
4. Check all queue
========================
1
Name: Angga
City origin: Solo
City Destination: Sidoarjo
Ticket Amount: 2
Price: 176000
Choose menu:
1. Queue
2. Dequeue
3. Check first queue
4. Check all queue
========================
1
Name: Fadin
City origin: Banyuwangi
City Destination: Bandung
Ticket Amount: 1
Price: 65000
Choose menu:
1. Queue
2. Dequeue
3. Check first queue
4. Check all queue
========================
3
The first element : Angga Solo Sidoarjo 176000 2
```

Questions

1. In Queue Class, what's the function of this program code in method Dequeue?

```
Passenger data = new Passenger("", "", "", 0, 0);
```

   - The purpose of the code in the Dequeue() method is to remove an element from the front of the queue and return the dequeued data.

2. In previous number, if the program code changed to

   Passenger data = new Passenger()

   What will happen?

   - Changing Passenger data = new Passenger() would create a new instance of Passenger with default values, not representing any actual data from the queue.

3. Show the program code used for displaying the data retrieved / removed from the queue!

```
case 2:
    Passengers data = queuePassenger.Dequeue();
    if (!"".equals(data.name) && !"".equals(data.cityOrigin) && !"".equals(data.cityDestination)
            && !"".equals(data.ticketAmount) && !"".equals(data.price)) {
        System.out.println("Data removed : " + data.name + " " + data.cityOrigin + " "
                + data.cityDestination + " " + data.ticketAmount + " " + data.price);
        break;
    }
```

```
    public Passengers Dequeue() {
        Passengers data = new Passengers(nm:"", co:"", cd:"", ta:0, pr:0);
        if (IsEmpty()) {
            System.out.println(x:"Queue is still empty");
        } else {
            data = Q[front];
            size--;
            if (IsEmpty()) {
                front = rear = -1;
            } else {
                if (front == max - 1) {
                    front = 0;
                } else {
                    front++;
                }
            }
        }
        return data;
    }
}
```

4. Modify the program by adding a method named peekRear() in Queue class to check the last position within the queue. Add a menu for the user to perform and explore your program as well

```
    public void peekRear() {
        if (!IsEmpty()) {
            System.out.println("The last element : " + Q[rear].name + " "
                    + Q[rear].cityOrigin + " " + Q[rear].cityDestination + " " +
                    Q[rear].ticketAmount + " " + Q[rear].price);
        } else {
            System.out.println(x:"Queue is still empty");
        }
    }
}
```

5. Ensure that the peekRear() function can be executed inside the program

```
    public static void menu() {
        System.out.println(x:"Choose menu: ");
        System.out.println(x:"1. Queue");
        System.out.println(x:"2. Dequeue");
        System.out.println(x:"3. Check first queue");
        System.out.println(x:"4. Check all queue");
        System.out.println(x:"5. Peek Rear");
        System.out.println(x:"==========================");
    }
```

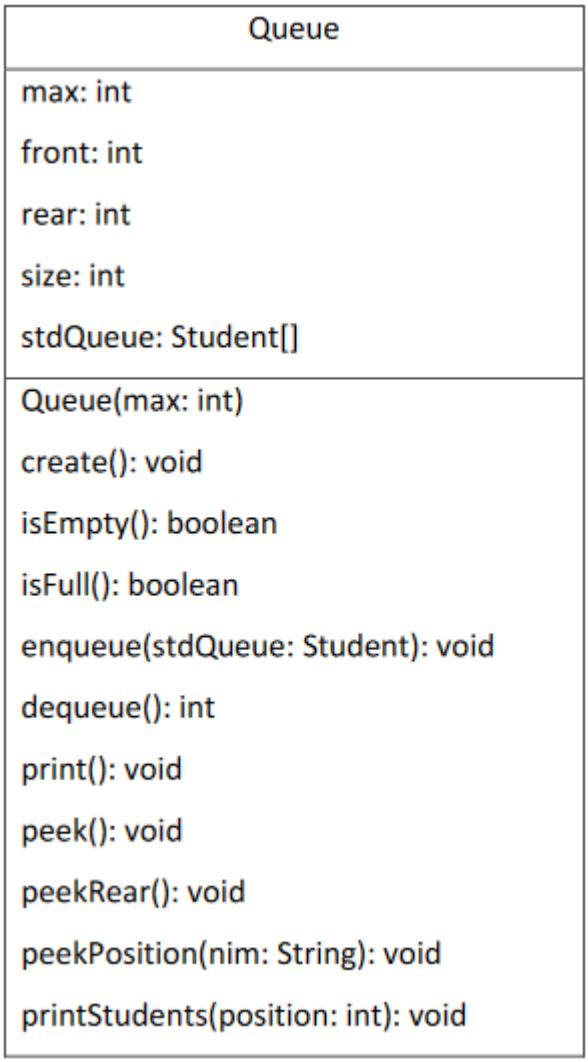
```
                case 5:
                    queuePassenger.peekRear();
                    break;
```

ASSIGNMENT

1. Add these 2 methods in Queue class in 1st practicum
2. Make a queue program for students when they need the signs for their KRS by the DPA. If the student is in queue, they will be required to fill in some information as follows:

Student
nim: String name: String classNumber: int gpa: double
Student (nim: String, name: String, classNumber: int,gpa: double)

Queue Class diagram:



Note:

- The implementation of Create(), isEmpty(), isFull(), enqueue(), dequeue() and print() functions are similar with we've built in practicum
- Peek() method is used for displaying students data in the first queue
- peekRead() method is used for displaying students data in the last queue
- peekPosition() method is used for displaying students data in the queue by their NIM
- printStudents() method is used for displaying a student data in specified position in a queue

SOURCE CODE

Queue,java: https://github.com/Garrss/Algoritma-Jobsheet/blob/main/Jobsheet10/Assigment/Queue.java

QueueMain.java: https://github.com/Garrss/Algoritma-Jobsheet/blob/main/Jobsheet10/Assigment/QueueMain.java

Student.java: https://github.com/Garrss/Algoritma-Jobsheet/blob/main/Jobsheet10/Assigment/Student.java

ANSWER

```
Enter the maximum number of students in the queue: 6

Menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Peek Rear
6. Peek Position
7. Print Student at Position
0. Exit
Enter your choice: 1
Enter NIM: 11001
Enter Name: nana
Enter Class Number: 2
Enter GPA: 3.55

Menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Peek Rear
6. Peek Position
7. Print Student at Position
0. Exit
Enter your choice: 1
Enter NIM: 11002
Enter Name: bobo
Enter Class Number: 2
Enter GPA: 3.44

Menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Peek Rear
6. Peek Position
7. Print Student at Position
```

```
6. Peek Position
7. Print Student at Position
0. Exit
Enter your choice: 1
Enter NIM: 11003
Enter Name: yoyo
Enter Class Number: 2
Enter GPA: 3.66

Menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Peek Rear
6. Peek Position
7. Print Student at Position
0. Exit
Enter your choice: 4
NIM: 11001, Name: nana, Class Number: 2, GPA: 3.55

Menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Peek Rear
6. Peek Position
7. Print Student at Position
0. Exit
Enter your choice: 5
NIM: 11003, Name: yoyo, Class Number: 2, GPA: 3.66

Menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Peek Rear
6. Peek Position
```

```
5. Peek Rear
6. Peek Position
7. Print Student at Position
0. Exit
Enter your choice: 6
Enter NIM to search: 11002
NIM: 11002, Name: bobo, Class Number: 2, GPA: 3.44

Menu:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Peek Rear
6. Peek Position
7. Print Student at Position
0. Exit
Enter your choice: 0
Exiting...
PS D:\Algoritma-Jobsheet>
```