

I Listen

1 Warteschlangen

Wo gibt es Warteschlangen?

Welche Unterschiede gibt es?

Wie entsteht eine Warteschlange?

Eine Warteschlange entsteht, wenn pro Zeiteinheit mehr Anforderungen an ein System gerichtet werden, als dieses verarbeiten kann.

Grundprinzip einer Warteschlange:

In einer Warteschlange (engl. queue) werden die Objekte in der gleichen Reihenfolge entnommen, wie sie eingefügt wurden.

FIFO-Prinzip: "first in first out"



1



2



3



4



5



6

Grundprinzip einer Warteschlange:

In einer Warteschlange (engl. queue) werden die Objekte in der gleichen Reihenfolge entnommen, wie sie eingefügt wurden.

FIFO-Prinzip: "first in first out"



1



2



3



4



5



6

Grundprinzip einer Warteschlange:

In einer Warteschlange (engl. queue) werden die Objekte in der gleichen Reihenfolge entnommen, wie sie eingefügt wurden.

FIFO-Prinzip: "first in first out"



1



2



3



4



5

6

Implementierung mit einem Feld:

Gib eine Klassenkarte der Klasse Warteschlange an!

WARTESCHLANGE	
→	anzahl
→	schlange
→	hintenEinfügen(objekt)
→	vorneEntnehmen()
	istLeer()
	alleAusgeben()
	anzahlGeben()

Das Aufrufen der Patienten im Wartezimmer von Dr. Heilmann soll in Zukunft entsprechend der Reihenfolge ihres Eintreffens mit Hilfe der EDV geregelt werden. Da es im Wartezimmer nur 12 Stühle gibt, müssen nur 12 Patienten verwaltet werden. Weitere Patienten werden einfach wieder nach Hause geschickt. Von den Patienten wird der Name und der Grund des Arztbesuchs eingetragen.

Entwickle die erweiterten Klassenkarten der Klassen Patient und Patientenwarteschlange.

Erweitert:

Attribute: + Zugriffsmodifikator und Datentyp

Methoden: ganze Signatur

PATIENT

```
private String name  
private String anlass
```

```
public PATIENT(String n, String a)  
public String nameGeben()  
public String anlassGeben()
```

PATIENTENWARTESCHLANGE

```
private int anzahlPatienten  
private Patient[] warteliste
```

```
public PATIENTENWARTESCHLANGE()  
public void hintenEinfügen(Patient p)  
public Patient vorneEntnehmen()  
public boolean istLeer()  
public void alleAusgeben()  
public int anzahlGeben()
```

```
public class Patient {  
    private String name;  
    private String anlass;  
  
    public Patient(String n, String a) {  
        name = n;  
        anlass = a;  
    }  
  
    public String nameGeben(){  
        return name;  
    }  
  
    public String anlassGeben(){  
        return anlass;  
    }  
}
```