

```
public class Patientenwarteschlange{  
    private int anzahlPatienten;  
    private Patient[] warteliste;  
  
    public Patientenwarteschlange()  
    {  
        anzahlPatienten = 0;  
        warteliste = new Patient[12];  
    }  
}
```

```
public void hintenEinfuegen(Patient p){  
    if
```

```
        else {
```

```
        }
```

```
    }
```

```
public void hintenEinfuegen(Patient p){  
    if (anzahlPatienten == 12){  
        System.out.println("Bitte nach Hause gehen!  
                             Kein Platz mehr!");  
    }  
    else {  
        hintenEinfuegen(p);  
    }  
}
```

```
public void hintenEinfuegen(Patient p){  
    if (anzahlPatienten == 12){  
        System.out.println("Bitte nach Hause gehen!  
                             Kein Platz mehr!");  
    }  
    else {  
        warteliste[anzahlPatienten] =  
        anzahlPatienten =  
    }  
}
```

```
public void hintenEinfuegen(Patient p){  
    if (anzahlPatienten == 12){  
        System.out.println("Bitte nach Hause gehen!  
                             Kein Platz mehr!");  
    }  
    else {  
        warteliste[anzahlPatienten] = p;  
        anzahlPatienten = anzahlPatienten + 1;  
    }  
}
```

Implementiere die Methode vorneEntnehmen(). Welcher Sonderfall muss berücksichtigt werden und was wird in diesem Fall zurückgegeben?

```
public Patient vorneEntnehmen(){
```

```
if {
```

}

```
else {
```

}

}

```
public Patient vorneEntnehmen(){
```

```
    if (anzahlPatienten == 0) {
```

```
        System.out.println("Fehler: Das Wartezimmer ist leer!");
```

```
    }
```

```
    else {
```

```
    }
```

```
}
```

```
public Patient vorneEntnehmen(){
    Patient erster = null;
    if (anzahlPatienten == 0) {
        System.out.println("Fehler: Das Wartezimmer ist leer!");
    }
    else {
        erster =

    }
    return erster;
}
```



```
public Patient vorneEntnehmen(){
    Patient erster = null;
    if (anzahlPatienten == 0) {
        System.out.println("Fehler: Das Wartezimmer ist leer!");
    }
    else {
        erster = warteliste[0];
        for (int i=0; i< warteliste.length; i++) {
            warteliste[i]=
                warteliste[i+1];
        }
        warteliste[warteliste.length-1] = null;
    }
    return erster;
}
```

```
public Patient vorneEntnehmen(){
    Patient erster = null;
    if (anzahlPatienten == 0) {
        System.out.println("Fehler: Das Wartezimmer ist leer!");
    }
    else {
        erster = warteliste[0];
        for (int i=0; i<anzahlPatienten-1; i++) {
            warteliste[i]=warteliste[i+1];
        }
        warteliste[anzahlPatienten-1] =
        anzahlPatienten =
    }
    return erster;
}
```

```
public Patient vorneEntnehmen(){
    Patient erster = null;
    if (anzahlPatienten == 0) {
        System.out.println("Fehler: Das Wartezimmer ist leer!");
    }
    else {
        erster = warteliste[0];
        for (int i=0; i<anzahlPatienten-1; i++) {
            warteliste[i]=warteliste[i+1];
        }
        warteliste[anzahlPatienten-1] = null;
        anzahlPatienten = anzahlPatienten - 1;
    }
    return erster;
}
```

```
public boolean istLeer(){  
    return (anzahlPatienten == 0);  
}
```

```
public void alleAusgeben(){  
    System.out.println("Warteliste:");  
    for(int i=0; i<anzahlPatienten; i++){  
        System.out.println(warteliste[i].nameGeben());  
    }  
}
```

```
public int anzahlGeben(){  
    return anzahlPatienten;  
}
```

Teste dein Programm. Du kannst dazu die Klasse Sprechstunde benutzen (Mebis).

AUTO

modell

farbe

inOrdnung

datenwerteGeben()

inOrdnungSetzen(io)

```
public class Auto {  
    private String modell;  
    private String farbe;  
    private boolean inOrdnung;
```

```
    public Auto(String m, String f) {  
        modell = m;  
        farbe = f;  
    }
```

```
    public String datenwerteGeben() {  
        return modell + ", " + farbe + ", " + inOrdnung;  
    }
```

```
    public void inOrdnungSetzen(boolean io) {  
        inOrdnung = io;  
    }  
}
```

AUTOWARTESCHLANGE

anzahlAutos
autoliste

hintenAnstellen(Auto)
vorneAbfahren()
istLeer()
istVoll()

```
public class Autowarteschlange {  
    private int anzahlAutos;  
    private Auto[] autoliste;  
  
    public Autowarteschlange() {  
        anzahlAutos = 0;  
        autoliste = new Auto[10];  
    }  
}
```

```
public void hintenAnstellen(Auto a){  
    if (anzahlAutos==10){  
        System.out.println("Fehler: Kein Platz mehr frei!");  
    }  
    else {  
        autoliste[anzahlAutos]=a;  
        anzahlAutos = anzahlAutos+1;  
    }  
}
```

Statt **anzahlAutos == 10** kann man auch
anzahlAutos == autoliste.length schreiben.
Das ist weniger fehleranfällig (z.B. bei Änderung der Feldlänge).


```
public Auto vorneAbfahren(){
    Auto erstesAuto= null;
    if (anzahlAutos == 0) {
        System.out.println("Fehler: Kein Auto in der Schlange!");
    }
    else {
        erstesAuto = autoliste[0];
        for (int i=0; i<anzahlAutos-1; i++) {
            autoliste[i]=autoliste[i+1];
        }
        autoliste[anzahlAutos-1]=null;
        anzahlAutos=anzahlAutos-1;
    }
    return erstesAuto;
}
```

```
public boolean istLeer() {  
    return (anzahlAutos==0);  
}
```

```
public boolean istVoll() {  
    return (anzahlAutos == 10);  
}
```

```
}
```

Erzeuge Objekte und teste dein Programm!

POLIZEIKONTROLLE

kontrollliste

autoHereinwinken(auto)

autoKontrollieren()

kontrolllisteGeben()

Bei der Kontrolle soll im Programm durch einen Zufallswert festgelegt werden, ob das Auto in Ordnung ist.

```
public class Polizeikontrolle {  
    private Autowarteschlange aws = new Autowarteschlange();  
    private String kontrollliste = "";  
  
    public void autoHereinwinken(Auto a) {  
        if  
        else  
    }  
}
```

POLIZEIKONTROLLE

kontrollliste

autoHereinwinken(auto)

autoKontrollieren()

kontrolllisteGeben()

Bei der Kontrolle soll im Programm durch einen Zufallswert festgelegt werden, ob das Auto in Ordnung ist.

```
public class Polizeikontrolle {  
    private Autowarteschlange aws = new Autowarteschlange();  
    private String kontrollliste = "";  
  
    public void autoHereinwinken(Auto a) {  
        if (aws.istVoll()) {System.out.println("Autowarteschlange voll!");}  
        else {aws.hintenAnstellen(a);}  
    }  
}
```

```
public void autoKontrollieren() {  
    if (aws.istLeer()) {  
        aws.add();  
    }  
    else {  
        aws.remove();  
    }  
}
```

```
public void autoKontrollieren() {  
    if (aws.istLeer()) {System.out.println("Autowarteschlange leer!");}  
    else {  
        Auto kontrollauto =  
            ;  
        if (Math.random()<0.5) kontrollauto.inOrdnungSetzen(true);  
        else kontrollauto.inOrdnungSetzen(false);  
        kontrollliste = kontrollliste  
            +"\n"  
            +  
    }  
}
```

Steuerzeichen:
\n neue Zeile
\t Tabulator
' Apostroph
" Gänsefüßchen

}

```
public void autoKontrollieren() {  
    if (aws.istLeer()) {System.out.println("Autowarteschlange leer!");}  
    else {  
        Auto kontrollauto = aws.vorneAbfahren();  
        if (Math.random()<0.5) kontrollauto.inOrdnungSetzen(true);  
        else kontrollauto.inOrdnungSetzen(false);  
        kontrollliste = kontrollliste  
                        +"\n"  
                        +kontrollauto.datenwerteGeben();  
    }  
}
```

```
public String kontrolllisteGeben() {  
    return kontrollliste;  
}  
}
```