```
/*
 * Verwaltet Patienten in einer Warteschlange
 * (Name, Datum)
 */
public class Wartezimmer
{
   private Queue<Patient> queue;

   public Wartezimmer()
   {
      queue  = new Queue();
   }

   /*
    * Patient wird in die Warteschlange eingefügt
    */
   public void einfuegen(Patient pPat)
   {
      queue.enqueue(pPat);
   }

   /*
    * Entfernt den Pat. am Anfang der Warteschlange und gibt ihn zurück
    */
   public Patient naechsterPatient()
   {
      Patient pat;

      if (queue.isEmpty())
      {
         return null;
      }
      else
      {
         pat = queue.front();
         queue.dequeue();
         return pat;
      }
   }

   /*
    * Alle Patienten werden aus der Warteschlange entfernt.
    */
   public void alleLoeschen()
   {
      while (!queue.isEmpty())
      {
         queue.dequeue();
      }
   }
```

```
/*
 * Gibt die Anzahl der in der Queue wartenden Patienten zurueck.
 */
public int anzahl()
{
   Patient erster, pat;
   int anzahl;

   if (queue.isEmpty())
   {
      return 0;
   }
   else
   {
      erster = queue.front();
      anzahl = 0;

      do {
         pat = queue.front();
         queue.dequeue();
         queue.enqueue(pat);
         anzahl++;
      }  while (queue.front() != erster);
   }
   return anzahl;
}

/*
 * Der Patient mit Namen "pName" wird aus der Warteschlange entfernt.
 * Vereinfachung: Namen kommen nicht mehrfach vor
 * Falls der Patient gelöscht wurde, wird true zurückgegeben
 */
public boolean loeschen(String pName)
{
   Patient pat, erster;
   boolean geloescht = false;

   // Sonderfall 1: Schlange leer
   if (queue.isEmpty())
   {
      return false;
   }
   else
   {
      // Sonderfall 2: der erste Patient ist der gesuchte
      if (queue.front().getName().equals(pName))
      {
         queue.dequeue();
         geloescht = true;
      }
      else
      {
         erster = queue.front();
         do
         {
            pat = queue.front();
            queue.dequeue();
```

```
                // Patient nur einfuegen, wenn nicht der zu loeschende
                if (pat.getName().equals(pName))
                {
                   geloescht = true;
                }
                else
                {
                   queue.enqueue(pat);
                }
             } while (queue.front() != erster);
          }
       }  // Queue nicht leer

       return geloescht;
    }
}


/*
 * Benutzeroberfläche für die Verwaltung wartender Patienten
 * (Name, Datum)
 */
public class WartezimmerUI
{
   private Wartezimmer wz;

   public WartezimmerUI()
   {
      wz = new Wartezimmer();
   }

   /*
    * Ein neuer Patient steht am Empfang.
    * Seine Daten werden über die Konsole eingegeben,
    * dann wird der Patient in die Warteschlange eingefügt.
    */
   public void anstellen()
   {
      String name, kasse;
      Patient pat;

      Console.println("Neuen Patient aufnehmen");
      Console.print("Wie ist der Name? ");
      name = Console.readln();
      Console.print("Welche Krankenkasse hat " + name + " ? ");
      kasse = Console.readln();

      pat = new Patient(name, kasse);
      wz.einfuegen(pat);

      Console.println(name + " ins Wartezimmer aufgenommen.");
   }
```

```
/*
 * Der naechste Patient kann vom Arzt behandelt werden.
 * Gibt den Patient am Anfang der Warteschlange aus entfernt ihn.
 */
public void aufrufen()
{
   Patient pat;

   Console.println("Patient aufrufen:");
   pat = wz.naechsterPatient();
   if (pat == null)
   {
      Console.println("Kein Patient in der Warteschlange.");
   }
   else
   {
      Console.println("Der naechste Patient ist " + pat.getName());
   }
}

/*
 * Gibt die Zahl der wartenden Patienten aus.
 */
public void status()
{
   int a;

   Console.println("Status des Wartezimmers:");
   a = wz.anzahl();
   if (a == 0)
   {
      Console.println("Das Wartezimmer ist leer.");
   }
   else
   {
      Console.println("Es warten " + a + " Patienten.");
   }
}

/*
 * Ein Patient wird vor der Behandlung aus der Warteschlange entfernt
 */
public void entfernen()
{
   String name;

   Console.println("Patient aus der Warteschlange entfernen:");
   Console.println("Geben Sie den Namen des Patienten ein.");
   name = Console.readln();

   if (wz.loeschen(name) == true)
   {
      Console.println("Patient " + name + " wurde entfernt.");
   }
```

```
      else
      {
         Console.println("Patient wurde nicht gefunden.");
      }
   }

   /*
    * Leert die Warteschlange
    */
   public void beenden()
   {
      wz.alleLoeschen();
      Console.println("Programm beendet.");
   }

   /*
    * Hauptmethode:
    * Zeigt ein Menü mit den Möglichkeiten des Programms an.
    * Benutzer wählen eine Möglichkeit aus, dann wiederholt es sich.
    */
   public void main()
   {
      int wahl;

      do
      {
         Console.clear();
         Console.println("=== Wartezimmer - Menue ===");
         Console.println("Bitte waehlen Sie:");
         Console.println("1: Neuen Patienten aufnehmen");
         Console.println("2: Patient zum Behandeln aufrufen");
         Console.println("3: Status des Wartezimmers");
         Console.println("4: Patient vorzeitig entlassen");
         Console.println("5: Programm beenden");
         Console.println();

         wahl = Console.readInt();
         Console.clear();

         if (wahl == 1)       { anstellen(); }
         else if (wahl == 2)  { aufrufen();  }
         else if (wahl == 3)  { status();    }
         else if (wahl == 4)  { entfernen(); }
         else if (wahl == 5)  { beenden();   }
         else                 { Console.println("Ungueltige Eingabe."); }
         Console.readln();

      } while (wahl != 5);
   }
}
```