



Übungen zu **Praktikum Grundlagen der Programmierung**

Aufgabe 37 Mensa-Theke (Lösungsvorschlag)

```
public class Person {
    private int priority;
    public Person() {
        setPriority(0);
    }
    protected void setPriority(int prio) {
        priority = prio;
    }
    public int getPriority() {
        return priority;
    }
}

public class Student extends Person {
    public Student() {
        setPriority(1);
    }
}

public class Mitarbeiter extends Person {
    public Mitarbeiter() {
        setPriority(2);
    }
}

public class Prof extends Mitarbeiter {
    public Prof() {
        setPriority(3);
    }
}

public class Element {
    private Person person;
    private Element next;

    public Element(Person p, Element n) {
        person = p;
        next = n;
    }

    public Person getPerson() {
        return person;
    }

    public void setPerson(Person p) {
        person = p;
    }

    public Element getNext() {
        return next;
    }
}
```

```

    }

    public void setNext(Element e) {
        next = e;
    }
}

public class MensaTheke{
    private Element waiting;

    public MensaTheke(){

    }

    public void anstellen(Person p) {
        Element el = waiting;
        if (el == null) {
            waiting = new Element(p,null);
        }else{
            while (el != null){
                if (el.getPerson().getPriority() >= p.getPriority()){
                    if (el.getNext() == null){
                        Element last = new Element(p,null);
                        el.setNext(last);
                        break;
                    }else{
                        el = el.getNext();
                    }
                }else{
                    Element shift = new Element(el.getPerson(),el.getNext());
                    el.setPerson(p);
                    el.setNext(shift);
                    break;
                }
            }
        }
    }

    public Person bedienen(){
        if (waiting == null){
            return null;
        }else{
            Person p = waiting.getPerson();
            waiting = waiting.getNext();
            return p;
        }
    }

    public String toString() {
        String result = "";
        for (Element elem = waiting; elem != null; elem = elem.getNext())
            result = result + "└┘" + elem.getPerson();
        return result;
    }

    public static void main(String[] args) {
        MensaTheke mensa = new MensaTheke();
        Prof seidl = new Prof();
        Prof schlichter = new Prof();
        Mitarbeiter gawlitza = new Mitarbeiter();
        Mitarbeiter flexeder = new Mitarbeiter();
        Student s1 = new Student();
        Student s2 = new Student();

        mensa.anstellen(s1);
        mensa.anstellen(flexeder);
        mensa.anstellen(s2);
        mensa.anstellen(seidl);
        mensa.anstellen(gawlitza);
        mensa.anstellen(schlichter);

        assert(mensa.bedienen() == seidl);
        assert(mensa.bedienen() == schlichter);
        assert(mensa.bedienen() == flexeder);
        assert(mensa.bedienen() == gawlitza);
        assert(mensa.bedienen() == s1);
        assert(mensa.bedienen() == s2);
    }
}

```