

Arbeitsauftrag – Mitarbeiter – Lösung

```
abstract class MITARBEITER{
    protected String name;
    protected boolean festangestellt;
    abstract double gehaltRechnen();
    String namenGeben(){
        return name;
    }
}

class ARBEITER extends MITARBEITER{
    double stundenlohn;
    int stunden;
    ARBEITER(String n, double stl, int s){
        stundenlohn = stl;
        stunden = s;
        name = n;
    }
    double gehaltRechnen(){
        return stundenlohn* stunden;
    }
}

class ANGESTELLT extends MITARBEITER{
    int grundgehalt;
    double prämiquote;
    ANGESTELLT(String n, int gg, double pp){
        name = n;
        grundgehalt = gg;
        prämiquote = pp;
    }
    double gehaltRechnen(){
        return grundgehalt*(1+prämiquote);
    }
}

class MANAGER extends MITARBEITER{
    MANAGER(String n){
        name = n;
    }
    double gehaltRechnen(){
        return 1000000;
    }
}
```

```
class ABRECHNUNG{
    MITARBEITER[] mliste;
    int anzahl;
    ABRECHNUNG(){
        mliste = new MITARBEITER[100];
        anzahl = 0;
    }
    void mitarbeiterDazu(MITARBEITER m){
        mliste[anzahl] = m;
        anzahl++;
    }
    void gesamtverdienst(){
        double gv=0;
        for (int i=0; i<anzahl; i++){
            gv = gv + mliste[i].gehaltRechnen();
            System.out.print(mliste[i].namenGeben());
            System.out.println(mliste[i].gehaltRechnen());
        }
        System.out.println("Gesamtgehalt: " + gv);
    }
}

class TEST{
    void testmethode(){
        ABRECHNUNG k = new ABRECHNUNG();
        ARBEITER a1 = new ARBEITER("Opp", 16.50, 200);
        ANGESTELLT b1 = new ANGESTELLT("Mül", 2500, 0.1);
        MANAGER c1 = new MANAGER("Hjk");
        k.mitarbeiterDazu(a1);
        k.mitarbeiterDazu(b1);
        k.mitarbeiterDazu(c1);
        k.gesamtverdienst();
    }
}
```