

```

package anhang.sonnensystem;

public enum Eigenschaft {
    BAHNRADIUS { public double wert(Himmelskoerper h) {
        return h.abstandZurSonne(); } },
    DURCHMESSER { public double wert(Himmelskoerper h) {
        return h.durchmesser(); } },
    FALLBESCHLEUNIGUNG { public double wert(Himmelskoerper h) {
        return h.fallBeschleunigung(); } },
    TEMPERATUR { public double wert(Himmelskoerper h) {
        return h.durchschnittsTemperatur(); } },
    ROTATIONS_PERIODE { public double wert(Himmelskoerper h) {
        return h.rotationsPeriode(); } },
    UMLAUFZEIT { public double wert(Himmelskoerper h) {
        return h.umlaufZeit(); } },
    MASSE { public double wert(Himmelskoerper h) { return h.masse(); } };
    public abstract double wert(Himmelskoerper h);
}

```

```

package anhang.sonnensystem;
import java.util.*;

public class Verwalter {
    private static Set<Himmelskoerper> mBekannt = new HashSet<>();
    public static void registriere(Himmelskoerper h) { mBekannt.add(h); }
    public static Set<Himmelskoerper> anmeldungen() { return mBekannt; }
    public static Map<Himmelskoerper, Himmelskoerper>
        klassenZu(Comparator<Himmelskoerper> c) {
        List<Himmelskoerper> liste = new ArrayList<>(mBekannt);
        Collections.sort(liste, c);
        Map<Himmelskoerper, Himmelskoerper> erg = new HashMap<>();
        Himmelskoerper aktuell = null;
        for (Himmelskoerper h : liste) {
            if (aktuell == null || c.compare(aktuell, h) < 0) aktuell = h;
            if (c.compare(aktuell, h) == 0) erg.put(h, aktuell);
            if (c.compare(aktuell, h) > 0) throw new InternalError();
        }
        return erg;
    }
}

```

```

package anhang.vergleiche;
import anhang.sonnensystem.*; import java.util.*;
public class Splitter implements Comparator<Himmelskoerper>{
    private final Eigenschaft mEigenschaft;
    private double mSchnitt;

```

```

private Splitter(Eigenschaft e, double schnitt) {
    mEigenschaft = e; mSchnitt = schnitt;
}
public int compare(Himmelskoerper h1, Himmelskoerper h2) {
    int d1 = (int) Math.signum(mEigenschaft.wert(h1) - mSchnitt);
    int d2 = (int) Math.signum(mEigenschaft.wert(h2) - mSchnitt);
    if (d1 * d2 > 0) return 0;
    if (d1 != 0) return d1;
    if (d2 != 0) return -d2;
    return 0;
}
public static Splitter untenOben(Eigenschaft e, double bei) {
    return new Splitter(e, bei);
}
public static Splitter untenOben(Eigenschaft e, Himmelskoerper h) {
    return new Splitter(e, e.wert(h));
}
public static Comparator<Himmelskoerper> obenUnten(Eigenschaft e,
    double bei) {
    return Collections.reverseOrder(untenOben(e, bei));
}
public static Comparator<Himmelskoerper> obenUnten(Eigenschaft e,
    Himmelskoerper h) {
    return Collections.reverseOrder(untenOben(e, h));
}
public final Splitter fuer(Himmelskoerper h) {
    mSchnitt = mEigenschaft.wert(h);
    return new Splitter(mEigenschaft, mSchnitt);
}
}

```

```

package anhang.vergleiche;
import java.util.*;

public class SekundaerVergleicher<T> implements Comparator<T> {
    private Comparator<T> mErst, mDann;
    public SekundaerVergleicher(Comparator<T> erst, Comparator<T> dann) {
        mErst = erst; mDann = dann;
    }
    public int compare(T t1, T t2) {
        int primaer = mErst.compare(t1, t2);
        if (primaer != 0) return primaer;
        return mDann.compare(t1, t2);
    }
}

```

```

package anhang.sonnensystem;

public interface Himmelskoerper {
    double abstandZurSonne(); // in Millionen km
    double durchmesser(); // in km
    double fallBeschleunigung(); // in m/s2
    double durchschnittsTemperatur(); // in Grad Celsius
    double rotationsPeriode(); // in Stunden
    double umlaufZeit(); // in Tagen
    double masse(); // in kg
}

```

---

```

package anhang.sonnensystem;
import static anhang.universum.Konstanten.*;

public enum Planet implements Himmelskoerper {
    MERKUR(58., 4880., 3.7, 167., 58. * TAG, 88., 3.3E23),
    VENUS(108., 12100., 8.9, 464., 243 * TAG, 224., 4.9E24),
    ERDE(150., 12700., 9.8, 15., TAG, JAHR, 6.E24),
    MARS(228., 6800., 3.7, -55., TAG, 687., 6.4E23),
    JUPITER(778., 143000., 24.8, -108., 10., 11. * JAHR, 1.9E27),
    SATURN(1430., 120500., 10.4, -139., 10., 29. * JAHR, 5.7E26),
    URANUS(2900., 51000., 8.9, -197., 17., 84. * JAHR, 8.7E25),
    NEPTUN(4500., 50000., 11.1, -201., 16., 165. * JAHR, 1.E26);
    private final double mAbstand, mDurchmesser, mFallBeschleunigung,
        mTemperatur, mPeriode, mUmlaufZeit, mMasse;
    private Planet(double a, double d, double g, double t,
        double p, double ut, double m) {
        mAbstand = a; mDurchmesser = d; mFallBeschleunigung = g;
        mTemperatur = t; mPeriode = p; mUmlaufZeit = ut; mMasse = m;
        Verwalter.registriere(this);
    }
    public double abstandZurSonne() { return mAbstand; }
    public double durchmesser() { return mDurchmesser; }
    public double fallBeschleunigung() { return mFallBeschleunigung; }
    public double durchschnittsTemperatur() { return mTemperatur; }
    public double rotationsPeriode() { return mPeriode; }
    public double umlaufZeit() { return mUmlaufZeit; }
    public double masse() { return mMasse; }
}

```

---

```

package anhang.sonnensystem;
import static anhang.universum.Konstanten.*;

public enum Sonne implements Himmelskoerper {
    DIE_EINZIGE;
}

```

```

private final double mDurchmesser = 1.4E6, mFallBeschleunigung = 274.,
    mTemperatur = 5500., mPeriode = 25. * TAG, mMasse = 2.E30;
private Sonne() { Verwalter.registriere(this); }
public double abstandZurSonne() { return 0.; }
public double durchmesser() { return mDurchmesser; }
public double fallBeschleunigung() { return mFallBeschleunigung; }
public double durchschnittsTemperatur() { return mTemperatur; }
public double rotationsPeriode() { return mPeriode; }
public double umlaufZeit() {
    throw new UnsupportedOperationException();
}
public double masse() { return mMasse; }
}

```

---

```

package anhang.sonnensystem;
import java.util.*;
import static anhang.sonnensystem.Eigenschaft.*;

public class Universell extends EnumMap<Eigenschaft, Double>
    implements Himmelskoerper {
    public Universell(Eigenschaft e, double w) {
        super(Eigenschaft.class);
        put(e, w); Verwalter.registriere(this);
    }
    public double abstandZurSonne() { return get(BAHNRADIUS); }
    public double durchmesser() { return get(DURCHMESSER); }
    public double fallBeschleunigung() { return get(FALLBESCHLEUNIGUNG); }
    public double durchschnittsTemperatur() { return get(TEMPERATUR); }
    public double rotationsPeriode() { return get(ROTATIONS_PERIODE); }
    public double umlaufZeit() { return get(UMLAUFZEIT); }
    public double masse() { return get(MASSE); }
}

```

---

```

package anhang.universum;

public class Konstanten {
    public final static int STUNDE = 60 * 60;
    public static final int TAG = 24;
    public static int JAHR = 365;
    public final int LICHT_GESCHWINDIGKEIT = 299_792_458;
    double LICHT_JAHR = LICHT_GESCHWINDIGKEIT * JAHR * TAG * STUNDE;
}

```

---