Java programming fundamentals: herhalingsoefeningen

Je maakt alle classes in een package `be.vdab`

Persoon

Je maakt een class Persoon met twee eigenschappen:
voornaam en familienaam.
Je kan deze eigenschappen aan de constructor meegeven.
De class bevat een method getNaam.
Deze geeft een String terug met de volledige naam (= voornaam + spatie + familienaam)

Probeer dit uit in een class met een public static void main(String[] args).

PolitiekePartij – Politieker

Je maakt een class PolitiekePartij. Deze heeft een naam die je meegeeft aan de constructor.

Erf van de class Persoon een class Politieker.
Deze heeft een extra eigenschap van het type PolitiekePartij.

Probeer dit uit in een class met een public static void main(String[] args).

GoedeDoel

Je maakt een class GoedeDoel. Een goed doel heeft een naam die je meegeeft aan de constructor.

Je kan op een goed doel een of meerdere keren

- Geld storten
- De naam opvragen
- Het totaal gestorte bedrag opvragen

Probeer dit uit in een class met een public static void main(String[] args).
De gebruiker tikt daarin bedragen tot hij nul tikt. Je toont daarna het totaal gestorte bedrag.

De bedragen zijn van het type BigDecimal.

Kost – Opbrengst

Je maakt een interface Kost met een method declaratie BigDecimal getKost().

Je maakt een interface Opbrengst met een method declaratie BigDecimal getOpbrengst().

Je maakt een class Gebouw. Deze heeft een eigenschap huurprijs.
De class implementeert de interface Kost: de kost van een gebouw is zijn huurprijs.

Je maakt een class Instructeur. Deze heeft drie eigenschappen: maandwedde, uurPrijsPerLes, aantalUrenLesAanWerknemers. De class implementeert de interface Kost: de kost van een instructeur is zijn maandwedde. De class implementeert ook de interface Opbrengst:
de opbrengst van een instructeur is uurPrijsPerLes x aantalUrenLesAanWerknemers.

Je maakt in een class met een public static void main(String[] args) objecten van het type Gebouw en Instructeur. Plaats deze objecten in een array van het type Kost.
Bereken de totale kost van alle elementen in deze array.

Palindroom

Je laat de gebruiker een woord intikken en jij zegt of dit woord een palindroom is.

Breuk

Je maakt een class breuk. Deze heeft als eigenschappen teller en noemer.
Je geeft deze eigenschappen mee aan de constructor.
Als de noemer 0 is, werp je een `IllegalArgumentException`.

De toString() method toont de teller, het teken / en de noemer.

Probeer dit uit in een class met een public static void main(String[] args)
waarin de gebruiker een teller en noemer tikt.
Je vangt de eventuele `IllegalArgumentException` op en toont dan een foutmelding.

Voorbeeldoplossingen

Persoon

```
package be.vdab;
class Persoon {
    private final String voornaam;
    private final String familienaam;
    public Persoon(String voornaam, String familienaam) {
        this.voornaam = voornaam;
        this.familienaam = familienaam;
    }
    public String getNaam() {
        return voornaam + ' ' + familienaam;
    }
}

public class PersoonProgramma {
    public static void main(String args[]) {
        System.out.println(new Persoon("Joe", "Dalton").getNaam());
    }
}
```

PolitiekePartij – Politieker

```
package be.vdab;
class PolitiekePartij {
    private final String naam;
    public PolitiekePartij(String naam) {
        this.naam = naam;
    }
    String getNaam() {
        return naam;
    }
}
class Politieker extends Persoon {
    private PolitiekePartij partij;
    public Politieker(String voornaam, String familienaam) {
        super(voornaam, familienaam);
    }
    void setPartij(PolitiekePartij partij) {
        this.partij = partij;
    }
    PolitiekePartij getPartij() {
        return partij;
    }
}
public class PolitiekProgramma {
    public static void main(String[] args) {
        PolitiekePartij partij = new PolitiekePartij("Fiesta");
        Politieker joe = new Politieker("Joe", "Dalton");
        System.out.println(joe.getNaam());
        joe.setPartij(partij);
        System.out.println(joe.getPartij().getNaam());
    }
}
```

GoedeDoel

```
package be.vdab;
import java.math.BigDecimal;
import java.util.Scanner;

class GoedeDoel {
    private final String naam;
    private BigDecimal saldo = BigDecimal.ZERO;
    public GoedeDoel(String naam) {
        this.naam = naam;
    }
    public void storten(BigDecimal bedrag) {
        saldo = saldo.add(bedrag);
    }
    public BigDecimal getSaldo() {
        return saldo;
    }
    public String getNaam() {
        return naam;
    }
}

public class GoedeDoelProgramma {
    public static void main(String[] args) {
        GoedeDoel doel = new GoedeDoel("CliniClowns");
        Scanner scanner = new Scanner(System.in);
        BigDecimal bedrag = scanner.nextBigDecimal();
        while (bedrag.compareTo(BigDecimal.ZERO) != 0) {
            doel.storten(bedrag);
            bedrag = scanner.nextBigDecimal();
        }
        System.out.println(doel.getNaam());
        System.out.println(doel.getSaldo());
    }
}
```

Kost – Opbrengst

```
package be.vdab;
import java.math.BigDecimal;

interface Kost {
    BigDecimal getKost();
}

interface Opbrengst {
    BigDecimal getOpbrengst();
}

class Gebouw implements Kost {
    private BigDecimal huurprijs;
    public BigDecimal getHuurprijs() {
        return huurprijs;
    }
    public void setHuurprijs(BigDecimal huurprijs) {
        this.huurprijs = huurprijs;
    }
    @Override
    public BigDecimal getKost() {
        return huurprijs;
    }
}

class Instructeur implements Kost, Opbrengst {
    private BigDecimal maandWedde;
    private BigDecimal uurPrijsPerLes;
    private int aantalUrenLesAanWerknemers;
    public BigDecimal getMaandWedde() {
        return maandWedde;
    }
    public void setMaandWedde(BigDecimal maandWedde) {
        this.maandWedde = maandWedde;
    }
    public BigDecimal getUurPrijsPerLes() {
        return uurPrijsPerLes;
    }
    public void setUurPrijsPerLes(BigDecimal uurPrijsPerLes) {
        this.uurPrijsPerLes = uurPrijsPerLes;
    }
    public int getAantalUrenLesAanWerknemers() {
        return aantalUrenLesAanWerknemers;
    }
    public void setAantalUrenLesAanWerknemers(int aantalUrenLesAanWerknemers) {
        this.aantalUrenLesAanWerknemers = aantalUrenLesAanWerknemers;
    }
    @Override
    public BigDecimal getKost() {
        return maandWedde;
    }
```

```
    @Override
    public BigDecimal getOpbrengst() {
        return uurPrijsPerLes.multiply(
          BigDecimal.valueOf(aantalUrenLesAanWerknemers));
    }
}

public class KostOpbrengstProgramma {
    public static void main(String[] args) {
        Gebouw gebouw = new Gebouw();
        gebouw.setHuurprijs(BigDecimal.valueOf(1000));
        Instructeur instructeur = new Instructeur();
        instructeur.setMaandWedde(BigDecimal.valueOf(2000));
        Kost[] kosten = new Kost[] {gebouw, instructeur};
        BigDecimal totaleKost = BigDecimal.ZERO;
        for (Kost kost : kosten) {
            totaleKost = totaleKost.add(kost.getKost());
        }
        System.out.println(totaleKost);
    }
}
```

Palindroom

```
package be.vdab;
import java.util.Scanner;

public class PalindroomProgramma {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String woord = scanner.next();
        System.out.println(
            new StringBuilder(woord).reverse().toString().equals(woord) ?
              "palindroom" : "geen palindroom");
    }
}
```

Breuk

```
package be.vdab;
import java.util.Scanner;

class Breuk {
    private final int teller;
    private final int noemer;
    public Breuk(int teller, int noemer) {
        if (noemer == 0) {
            throw new IllegalArgumentException("noemer mag niet 0 zijn");
        }
        this.teller = teller;
        this.noemer = noemer;
    }
    @Override
    public String toString() {
        return teller + "/" + noemer;
    }
}

public class BreukProgramma {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int teller = scanner.nextInt();
        try {
            int noemer = scanner.nextInt();
            System.out.println(new Breuk(teller, noemer));
        } catch (IllegalArgumentException ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```