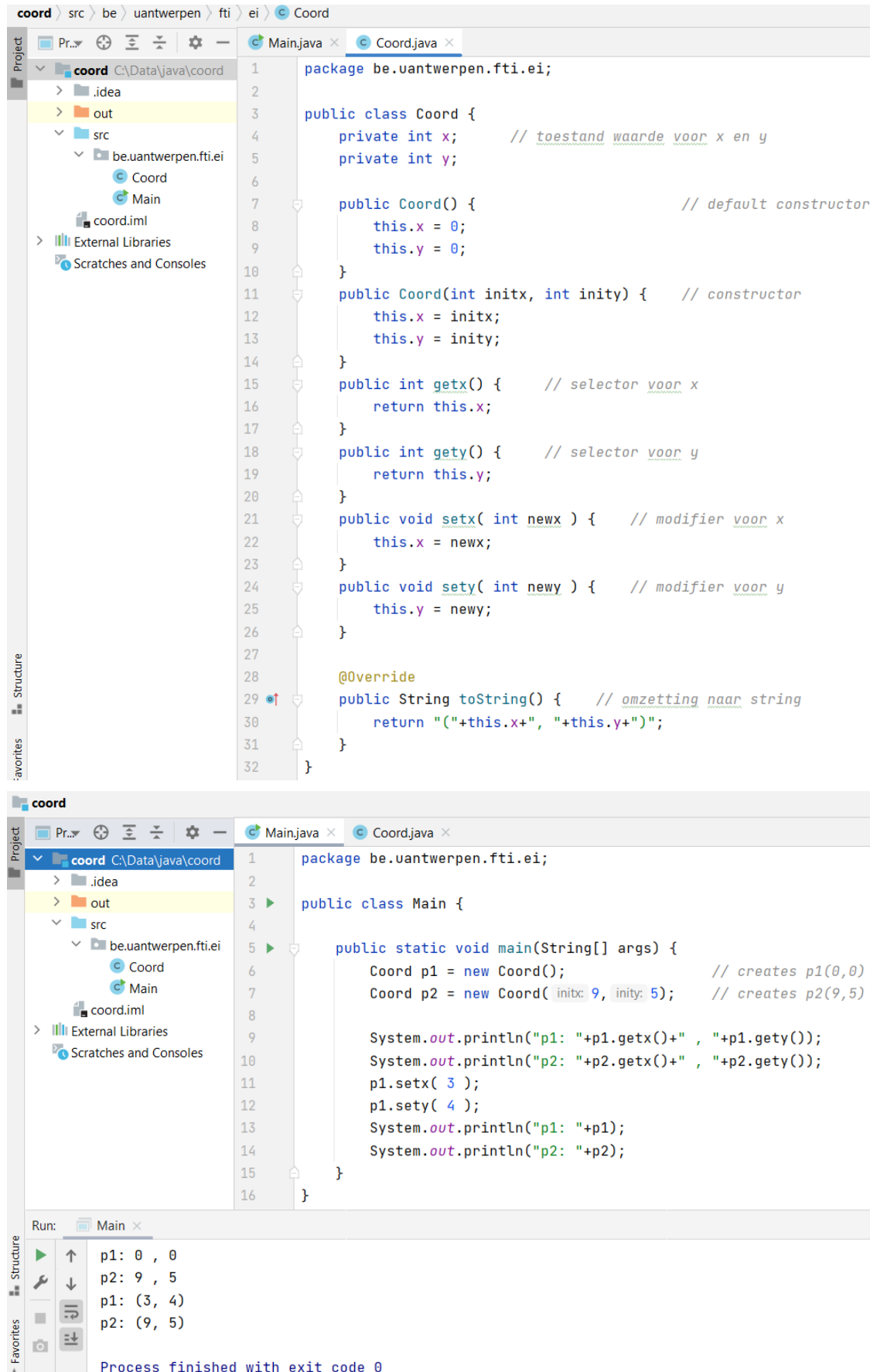


Oefeningen introductie Java

Voorbeeld met **Cartesische coördinaten**:



The screenshot displays an IDE with two Java files: `Coord.java` and `Main.java`.

Coord.java defines a `Coord` class with the following code:

```

1  package be.uantwerpen.fti.ei;
2
3  public class Coord {
4      private int x;      // toestand waarde voor x en y
5      private int y;
6
7      public Coord() {    // default constructor
8          this.x = 0;
9          this.y = 0;
10     }
11     public Coord(int initx, int inity) { // constructor
12         this.x = initx;
13         this.y = inity;
14     }
15     public int getX() { // selector voor x
16         return this.x;
17     }
18     public int getY() { // selector voor y
19         return this.y;
20     }
21     public void setx( int newx ) { // modifier voor x
22         this.x = newx;
23     }
24     public void sety( int newy ) { // modifier voor y
25         this.y = newy;
26     }
27
28     @Override
29     public String toString() { // omzetting naar string
30         return ("+"+this.x+", "+"+this.y+"");
31     }
32 }
    
```

Main.java defines a `Main` class with the following code:

```

1  package be.uantwerpen.fti.ei;
2
3  public class Main {
4
5      public static void main(String[] args) {
6          Coord p1 = new Coord(); // creates p1(0,0)
7          Coord p2 = new Coord( 9, 5); // creates p2(9,5)
8
9          System.out.println("p1: "+p1.getX()+" , "+p1.getY());
10         System.out.println("p2: "+p2.getX()+" , "+p2.getY());
11         p1.setx( 3 );
12         p1.sety( 4 );
13         System.out.println("p1: "+p1);
14         System.out.println("p2: "+p2);
15     }
16 }
    
```

The **Run** panel shows the output of the program:

```

p1: 0 , 0
p2: 9 , 5
p1: (3, 4)
p2: (9, 5)

Process finished with exit code 0
    
```

1. Schrijf een Java programma dat een klasse **Complex** gebruikt om een complex getal voor te stellen. Het reëel deel (**re**) en het imaginair deel (**im**) moeten bijgehouden worden als **double**.

Verder moeten methods voorzien worden om:

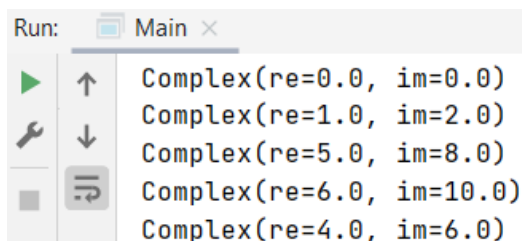
- de gegevens te initialiseren (2 **constructors**: default (0.0,0.0) en met re,im)
- de gegevens te wijzigen (**set**)
- de gegevens individueel op te vragen als getal (**get**)
- de som te maken van twee complexe getallen (**add**)
- het verschil te maken van twee complexe getallen (**subtract**)
- de gegevens terug te geven als string ("Complex(re, im)") (**toString**())

Het hoofdprogramma moet een demonstratie geven van het gebruik van alle methods.

Bijvoorbeeld: (test ook de get-methods)

```

Complex c1 = new Complex();
Complex c2 = new Complex(1.0, 2.0);
System.out.println(c1);
System.out.println(c2);
c1.setRe(5);
c1.setIm(8);
System.out.println(c1);
System.out.println(c1.add(c2));
System.out.println(c1.subtract(c2));
  
```



```

Run: Main x
Complex(re=0.0, im=0.0)
Complex(re=1.0, im=2.0)
Complex(re=5.0, im=8.0)
Complex(re=6.0, im=10.0)
Complex(re=4.0, im=6.0)
  
```

2. Schrijf een Java programma dat een class gebruikt om een datum vast te leggen.

Hierbij moet voor de **Datum** klasse (**dag**, **maand**, **jaar**) bijgehouden worden (3 x **int**).

Verder moeten methods voorzien worden om:

- de gegevens te initialiseren (constructors)
- de gegevens te wijzigen (set)
- de gegevens individueel op te vragen als getal (dag, maand en jaar)
- de gegevens terug te geven als string (toString())
- naar de volgende dag te gaan (method next of volgende)
- de gegevens te controleren op correctheid (vb. dagen per maand)

Let op voor schrikkeljaren!

- tijdstippen te vergelijken (interface Comparable, method compareTo)
- de huidige dag op halen in default constructor
- gebruik **exceptions** om een foutieve datum op te vangen in de constructor en de set methods. Creëer hiervoor een exception class “OngeldigeDatum” (InvalidDate).

Het hoofdprogramma moet een demonstratie geven van het gebruik van alle methods.

Maak bijvoorbeeld een ArrayList van Datum objecten aan en sorteer ze.

```

package be.uantwerpen.fti.ei;

public class Main {

    public static void main(String[] args) {
        try {
            Datum d1 = new Datum( dag: 28, maand: 2, jaar: 2022);
            Datum d2 = new Datum();
            System.out.println(d1);
            System.out.println(d2);
            d1.volgende();
            System.out.println(d1);
            d1.setDag(32);
        } catch (OngeldigeDatumException e) {
            e.printStackTrace();
        }
    }
}

```

```

Datum(dag=28, maand=2, jaar=2022)
Datum(dag=15, maand=2, jaar=2022)
Datum(dag=1, maand=3, jaar=2022)
be.uantwerpen.fti.ei.OngeldigeDatumException Create breakpoint : Ongeldige datum in setDag! (32)
    at be.uantwerpen.fti.ei.Datum.setDag(Datum.java:30)
    at be.uantwerpen.fti.ei.Main.main(Main.java:13)
Process finished with exit code 0

```

3. Schrijf een Java programma voor de volgende opgave:

Maak een class om een **Persoon** voor te stellen met als data members (private):

- String naam;
- String gebdat; // geboortedatum "dd/mm/jjjj"
- String adres;

Voorzie alle nodige methods. (get, set, constructors (default, init) , toString)

Maak een tweede class **Student** afgeleid van *Persoon* met als extra data members (private):

- String klas;
- double punten; // resultaat in %

Voorzie alle nodige functies. (get, set, constructors (default, init) , toString)

Maak een derde class **Kotstudent** afgeleid (public) van *Student* met als extra data member (private):

- String kotadres;

Voorzie alle nodige functies. (get, set, constructors (default, init) , toString)

Elke class krijgt een eigen functie om zijn gegevens om te zetten naar string (toString).

Zorg ervoor dat in alle gevallen de juiste gegevens netjes op het scherm worden weergegeven.

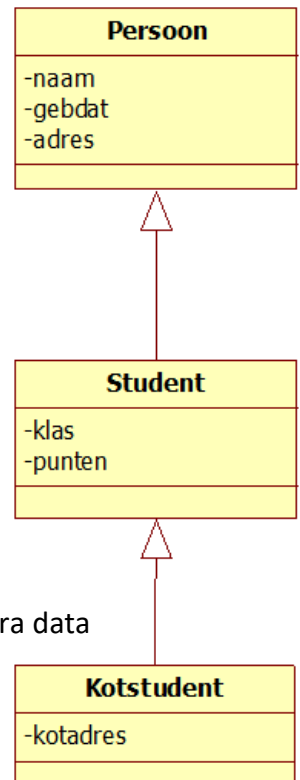
Demonstreer het gebruik van deze drie classes in het hoofdprogramma.

- 1 Begin met *Persoon*;
- 2 Voeg daarna *Student* toe;
- 3 Voeg *Kotstudent* toe als de eerste twee klassen in orde zijn en werken.

Het hoofdprogramma moet een demonstratie geven van het gebruik van alle methods.

Gebruik hierbij de werking van het polymorfiemechanisme bij het implementeren van toString.

Pas eventueel de "gebdat" in de klasse *Persoon* aan naar *Datum* i.p.v. *String*.



```

package be.uantwerpen.fti.ei;

public class Main {

    public static void main(String[] args) {
        Persoon p1 = new Persoon( naam: "Jan", gebdat: "5/7/2001", adres: "Bomenlaan 19");
        Persoon p2 = new Student( naam: "Ann", gebdat: "7/9/2000", adres: "Perendreef 3", klas: "2EI", punten: 15.0);
        Persoon p3 = new Kotstudent( naam: "Jef", gebdat: "3/5/2002", adres: "Thuisadres 8", klas: "2EI", punten: 13,
                                   kotadres: "Kotadres 872");

        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p3);
    }
}
  
```

Run: Main

```

Persoon(naam='Jan', gebdat='5/7/2001', adres='Bomenlaan 19')
Student(Persoon(naam='Ann', gebdat='7/9/2000', adres='Perendreef 3'), klas='2EI', punten=15.0)
Kotstudent(Student(Persoon(naam='Jef', gebdat='3/5/2002', adres='Thuisadres 8'), klas='2EI', punten=13.0),
kotadres='Kotadres 872')
  
```

Build completed successfully in 2 sec, 159 ms (4 minutes ago)

4. Maak een kopie van het project uit de vorige opgave en pas de volgende zaken aan:

Pas in de class “**Student**” de data member ‘punten’ aan naar een ArrayList van 3 punten. Deze class bevat nu de volgende data members:

- ‘klas’ als string
- ‘punten’ op 20 voor 3 vakken in een ArrayList van int

Pas alle methods aan zodat de 3 punten verwerkt worden:

- de gegevens van de student initialiseren (constructor met extra parameters voor de 3 punten i.p.v. 1)
- de gegevens van de student om te zetten naar string (inclusief de 3 punten)
- de get en set en methods voor punten (getPunten en setPunten) met een extra parameter vak (met als mogelijke waarde 0, 1 of 2) (punten per vak op te vragen en aan te passen)
- extra method voor het gemiddelde van de 3 punten te berekenen (gemiddelde)

Demonstreer het gebruik van deze aangepaste class in het hoofdprogramma.

Maak een ArrayList van Student objecten en **sorteer** ze volgens hun **gemiddelde punten van hoog naar laag**. Gebruik hiervoor **niet** de Comparable interface maar de **Comparator interface**.

Bijvoorbeeld (basis):

```
Student st = new Student("Jan", "05/07/2004", "Adres", "2EI", 13, 15, 12);
```

```
st.setPunten(0, 15);    // punten voor vak 0 op 15
```

```
System.out.println(st);
```

```
System.out.println("Het gemiddelde is " + st.gemiddelde());
```