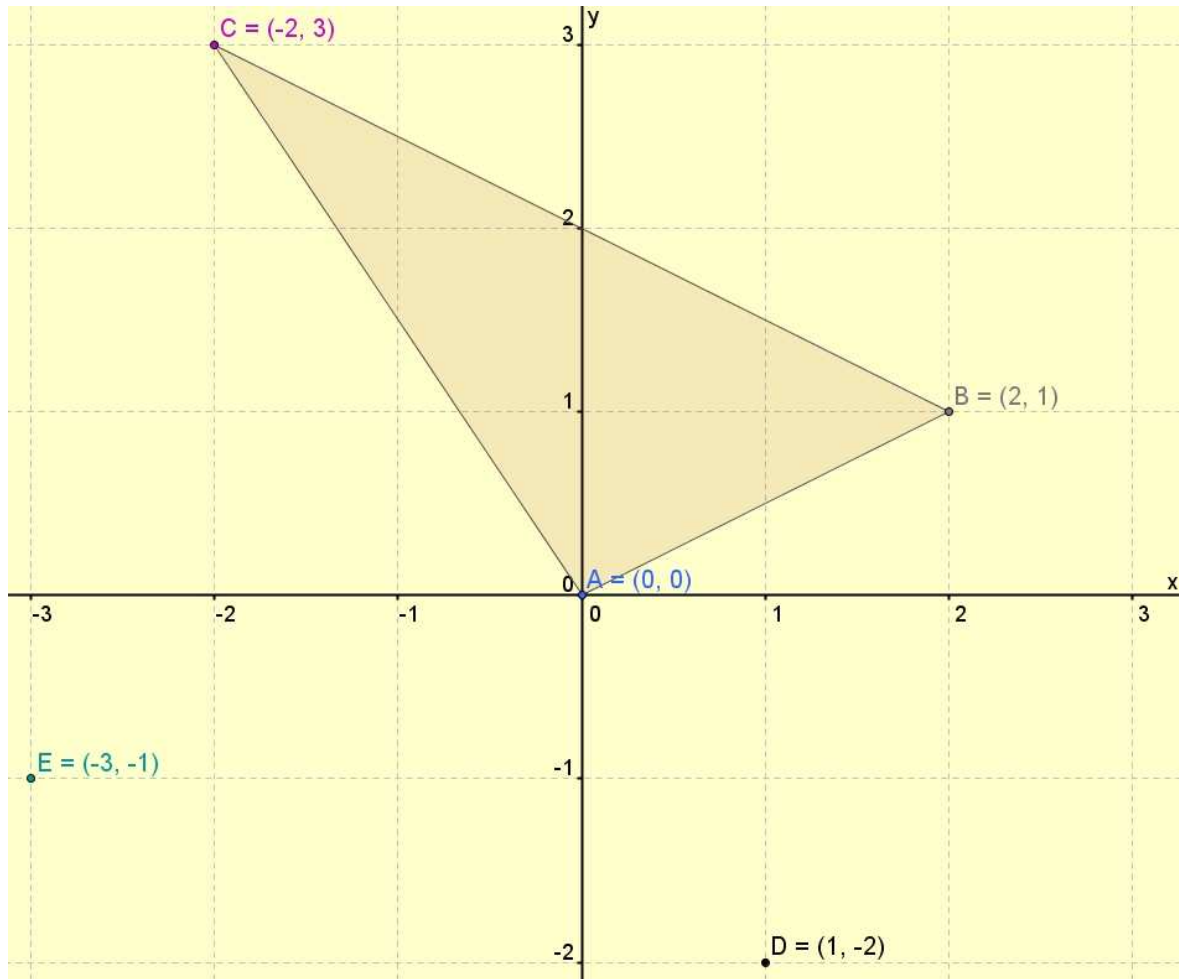


PP Practicum 4 – Punt en Veelhoek

Beschrijving

Een **punt** A wordt weergegeven door twee getallen, de x - en de y -coördinaat: (a_x, a_y) .

In de tekening zie je als voorbeeld vijf punten met hun coördinaten.



Een **veelhoek** is gedefinieerd als een array van (drie of meer) objecten van het type Punt. Zo is een veelhoek die uit drie punten bestaat natuurlijk een driehoek. In de tekening zie je bijvoorbeeld de driehoek die uit de punten A , B en C bestaat.

In deze opdracht maak je de klassen Punt en Veelhoek. Deze klassen hebben een aantal methoden. De applicatie moet worden getest vanuit een Main klasse.

Om het je gemakkelijker te maken is de opdracht in vijf stappen onderverdeeld.

Vorbereiding

Zorg voor een werkomgeving met de laatste versie van NetBeans.

Gebruik Subversion om versiebeheer met je source code te kunnen doen.

Voeg commentaar toe waar nodig, en let op naamgeving en layout

Opdracht A:

Class, constructor, method

- Maak een klasse **Punt** met de eigenschappen die in de beschrijving zijn genoemd (een Punt is gedefinieerd door zijn x - en y -coördinaat).
- Maak twee *constructors* voor de klasse **Punt**:
 - Een constructor zonder argumenten; deze zet het punt op (0, 0).
 - Een constructor met twee argumenten: de x - en y -coördinaat; deze zet het punt op de x - en y -coördinaat.
- Maak in de klasse **Punt** een methode *print()* die de coördinaten van het punt afdruckt.
- Test je klasse door in de *main()* methode van klasse Main twee objecten van het type Punt aan te maken (gebruik beide constructors!) en roep van beide objecten de *print()* methode aan.

Opdracht B:

Method

- Maak in de klasse **Punt** een methode *verschuif(dx, dy)*. Deze methode verschuift een punt een aantal stappen (dx) in de x -richting (positief is naar rechts, negatief is naar links) en een aantal stappen (dy) in de y -richting (positief is naar boven, negatief is naar beneden). Doe deze verschuiving door de eigenschappen van het punt aan te passen.
Voorbeeld: als je punt C(-2,3) verschuift met (3,-1), dan wordt de nieuwe x -coördinaat $-2+3=1$ en de nieuwe y -coördinaat $3-1=2$, dus het nieuwe punt C wordt (1,2).
- Test deze methode door in de *main()* methode van klasse Main het volgende te doen.
 - Maak een punt aan en druk zijn coördinaten af.
 - Verplaats het punt met de methode *verschuif()*.
 - Druk de nieuwe coördinaten van het verplaatste punt af.

Opdracht C:

Class, constructor, method

- Maak een klasse **Veelhoek** met de eigenschappen die in de beschrijving zijn genoemd (een Veelhoek bestaat uit een array van punten).
- Maak twee *constructors* voor de klasse **Veelhoek**:
 - Een constructor zonder argumenten; deze initialiseert een array van grootte drie en initialiseert de punten in het array allemaal op (0, 0).
 - Een constructor met één argument: de dimensie, dit is het aantal hoeken/punten waaruit de veelhoek bestaat; deze initialiseert een array van grootte van het gevraagde aantal punten en initialiseert de punten in het array allemaal op (0, 0).
Controle: Let op dat de dimensie drie of hoger moet zijn. Zo niet, dan moet er een passende foutmelding op het scherm worden afgedrukt en wordt vervolgens een driehoek gecreëerd.
- Maak in de klasse **Veelhoek** een methode *print()* die de dimensie van de veelhoek afdruckt, gevolgd door de coördinaten van ieder punt. Maak daarbij gebruik van de *print()* methode van de klasse Punt.
- Test je klasse door in de *main()* methode van klasse Main twee objecten van het type Veelhoek aan te maken (gebruik beide constructors!) en roep van beide objecten de *print()* methode aan.

Opdracht D:

Method

- Maak in de klasse **Veelhoek** een methode *setPunt(volgNummer, p)* om de punten van de veelhoek in te vullen. Deze methode krijgt twee argumenten mee:
 - Het volgnummer van het punt (bij een vijfhoek kan dat dus 1, 2, 3, 4 of 5 zijn)
 - Een object van het type Punt. Hiermee wordt het bewuste punt geïnitieerd.
 - **Controle:** Het volgnummer moet groter zijn dan 0 en kleiner of gelijk zijn dan de dimensie van de veelhoek. Zo niet, dan moet een passende foutmelding op het scherm worden afgedrukt.
- Test deze methode door in de *main()* methode van klasse Main de driehoek ABC uit het plaatje van de beschrijving te maken. Roep de *print()* method van die driehoek aan ter controle.

Opdracht E:

Method

- Maak in de klasse **Veelhoek** een methode *verschuif(dx, dy)*. Deze methode verschuift alle punten van de veelhoek een aantal stappen in de *x*-richting (*dx*) en een aantal stappen in de *y*-richting (*dy*). Maak daarbij gebruik van de *verschuif(dx,dy)* methode van de klasse Punt
- Test deze methode door in de *main()* methode van klasse Main het volgende te doen.
 - Maak een veelhoek aan en druk zijn punten af.
 - Verplaats de veelhoek met de methode *verschuif()*.
 - Druk de nieuwe punten van de verplaatste veelhoek af.