

Werk de oefening 7 af tegen dinsdag 31-10

Deze oefening is een mooie voorbereiding voor de gequoteerde oefening omdat we de leerstof van H2 en H3 nodig hebben.

Gegeven: startproject H04start_Tankstation

Opdracht 1

Werk de klasse Tankstation verder af. (m.a.w. de code aanvullen in de klasse Tankstation van je startproject, we tekenen de UML niet over in Visual Paradigm)

Gegeven (zie opgave oefening 7):

Tankstation
-pompen : int[]
+Tankstation() +tank(pompNummer : int, aantalLiter : int) : boolean +geefInhoud(pompNummer : int) : int +toString() : String

Voorzie 3 constanten, we gaan ze public zetten omdat de klasse TankbeurtApp van package cui gebruik zal maken van deze constanten.

```
public static final int MAX_POMPEN = 5;  
public static final int MIN_LITER = 5;  
public static final int MAX_LITER = 80;
```

```
public Tankstation() {  
    // TODO  
}
```

In deze constructor de array “pompen” creëren (lengte = MAX_POMPEN). Iedere waarde opvullen met 100.

100	100	100	100	100
0	1	2	3	4

```
public boolean tank(int pompNummer, int aantalLiter) {  
    // TODO  
}  
  
public int geefInhoud(int pompNummer) {  
    // TODO  
}
```

Methode tank: pompnummer moet in [1,5] liggen, aantalLiter moet in [5,80] liggen, er moet voldoende benzine zijn.

Bv: we roepen de methode tank(2, 70) op dan geeft de methode **false** terug (pompNummer 2 bevat slechts 60 liter benzine (let erop: java telt vanaf 0, dus pompNummer 2 komt overeen met index 1)

100	60	100	100	100
0	1	2	3	4

Methode geefInhoud: geeft het aantal liters terug van de gevraagde pompnummer

Bv. (hetzelfde voorbeeld als hierboven): we roepen de methode geefInhoud(2) op dan krijgen we **60** terug.

Klaar met implementeren van deze klasse, voer de klasse TankstationTest uit in je project

```
▼ [icon] domein
  > [icon] Tankstation.java
▼ [icon] testen
  > [icon] TankstationTest.java
```



Opdracht 2: Werk de klasse TankstationApp (package cui) af

Kijk naar de opdracht en haal de hoofdstructuur eruit.

```
Aan welke pomp wil je tanken (1-5), druk 0 om te stoppen?
```

Je dient 3 methoden uit te werken:

Methode startTanken: bevat de hoofdstructuur. Een object van tankStation creëren. We dienen “totaalLiterGetankt” en “aantalSuccesvolleTankBeurten” te bepalen.

Per tankbeurt (**dus in de lus**) toon je “Je hebt X liter getankt aan pomp Y”, bv.

```
Je hebt 25 liter getankt aan pomp 3
```

of “Sorry, er is onvoldoende brandstof aan pomp Z”, bv.

```
Sorry, er is onvoldoende brandstof aan pomp 1
```

Vervolgens toon je het overzicht, door de methode toString op te roepen van Tankstation.

Na de lus (m.a.w. de gebruiker geeft 0 ingegeven) toon je “Er werd in totaal X liter getankt in Y tankbeurten”, vb:

```
Er werd in totaal 150 liter getankt in 2 tankbeurten.
```

of

```
Er waren geen tankbeurten:
Pomp #1: 100 liter
Pomp #2: 100 liter
Pomp #3: 100 liter
Pomp #4: 100 liter
Pomp #5: 100 liter
```

Methode geefAantalLiter: aantal liters vragen. Indien foutief, foutboodschap tonen en opnieuw vragen, totdat de gebruiker een juiste waarde heeft ingegeven. De toegelaten min. en max. kan je als volgt opvragen:

Tankstation.[MIN_LITER](#) en Tankstation.[MAX_LITER](#) (m.a.w. we maken gebruik van de constanten van de klasse Tankstation)

```
Hoeveel liter wens je te tanken (5 tot 80)? 4
Je moet minstens 5 en kan maximaal 80 liter tanken...
Hoeveel liter wens je te tanken (5 tot 80)? 90
Je moet minstens 5 en kan maximaal 80 liter tanken...
Hoeveel liter wens je te tanken (5 tot 80)? 80
```

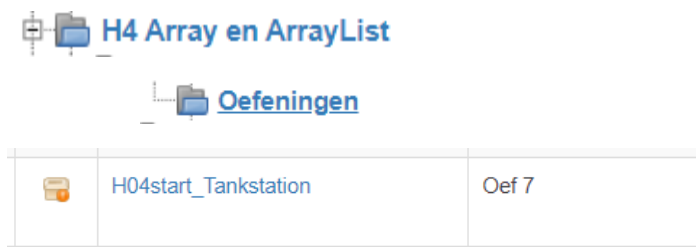
Methode geefPompNummer: vraagt aan de gebruiker aan welk pomp je wil tanken of 0 om te stoppen. Indien ongeldige invoer, foutboodschap tonen en invoer opnieuw vragen totdat de gebruiker juiste invoer ingeeft? Maak gebruik van de constante Tankstation.**MAX_POMPEN:**

```
geldigeInvoer = ??? pompNummer <= Tankstation.MAX_POMPEN;
```

```
Aan welke pomp wil je tanken (1-5), druk 0 om te stoppen? 6
We hebben geen pomp met nummer 6! Probeer opnieuw...
Aan welke pomp wil je tanken (1-5), druk 0 om te stoppen? -5
We hebben geen pomp met nummer -5! Probeer opnieuw...
Aan welke pomp wil je tanken (1-5), druk 0 om te stoppen? 3
```

Voor de studenten die niet aanwezig waren in de les van vrijdag 27-10:

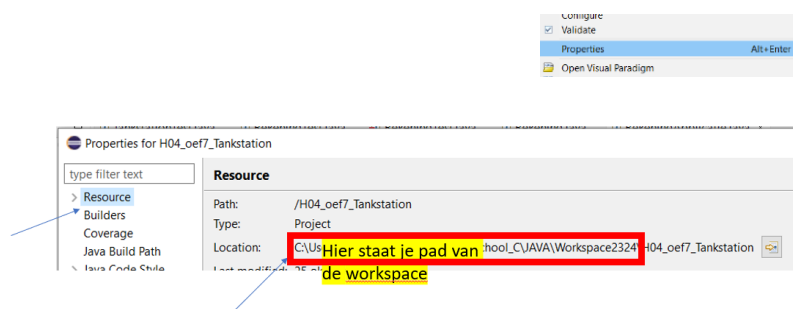
Haal het startproject van chamilo:



Startproject downloaden en bewaren in de map van je workspace van Eclipse.

(Indien je de workspace van je Eclipse niet kent:

Rechtermuisklik op een project in je Eclipse: kies Properties



)

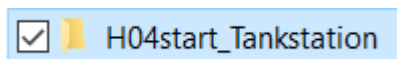
In de workspace staat:

H04start_Tankstation	27/10/2023 10:27	Gecomprimeerde ...	8 kB
----------------------	------------------	--------------------	------

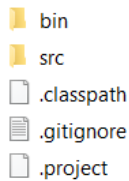
De zip uitpakken (hier uitpakken, zonder een voorgestelde map “H04start_Tankstation”).

Nadat het uitgepakt is hebben we **1** map “H04start_Tankstation” met daarin bin, src..., **niet 2** mappen

In de map van je workspace staat:

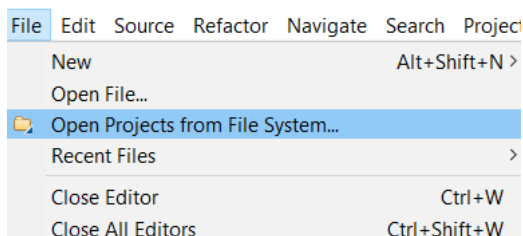


Doe de map open en je dient bin, src, ... te zien:

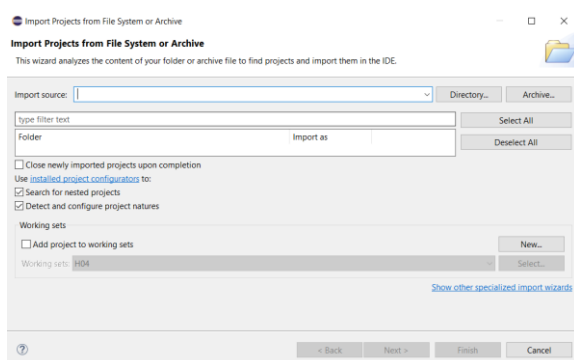


In Eclipse, sta op je working set van H4.

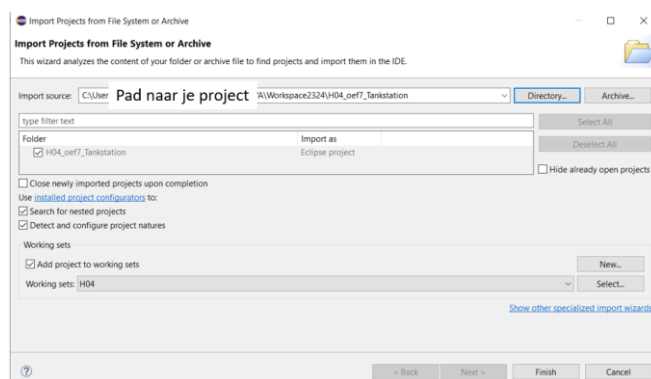
Ga naar menu “File”, “Open Projects from File System...”



Je krijgt een nieuw scherm:



Met de knop “Directory...” navigeer je naar je project.



Knop “Finish”. Project staat nu in je working set van H4.

Veel succes en tot dinsdag