

---

# PROEFTEST 1 SOFTWAREONTWIKKELING I

1 maart 2016

NAAM:

COMPUTER:

---

De opgave van de test (opgave.pdf) vind je in de map `c:\temp\opgave`. In deze test schrijf je verschillende klassen om krijgers aan te maken, voor te stellen en te sorteren.

## Documentatie

Je vindt de Java API op het bureaublad.

## Oplossing

In de map `c:\temp\oplossing` vind je een bestand `naam.txt`. Schrijf in dit bestand je naam en voornaam en bewaar het bestand. In deze map vind je ook een Netbeans-project **Krijgers**, waarin jouw oplossing van de test moeten komen. Een kopie van dit project vind je ook in de map `c:\temp\opgave` voor het geval je per ongeluk iets verwijdt.

Op het einde van de test:

- Zorg ervoor dat de code compileerbaar en (zoveel mogelijk) uitvoerbaar is.
- Alles bewaren.
- Voeg eventuele opmerkingen in commentaar toe (bv. niet uitgewerkte ideeën, ...)
- Netbeans afsluiten.
- Uitloggen, maar de computer niet afzetten.
- Opgaveblad afgeven.

## Opdracht

1. Schrijf de klassen om krijgers voor te stellen. Maak hierbij gebruik van overerving. Zorg ervoor dat je geen code dupliceert. Let ook op de toegankelijkheid van attributen en methodes.

Krijger-objecten hebben de volgende kenmerken.

- De eigenschappen van een krijger zijn: zijn naam, het aantal levens waarover hij beschikt, het aantal gewonnen gevechten, zijn aanvalskracht en zijn verdedigingskracht.
- De aanvals- en verdedigingskracht zijn afhankelijk van het type krijger. Er zijn twee krijgertypes: ninja's en samoerai. De krachten worden bij elke aanroep opnieuw berekend.
  - Indien een ninja nog  $i$  levens heeft en al  $j$  gevechten gewonnen heeft, is zijn aanvalskracht een willekeurig getal gelegen tussen 0 en  $(80 * i + 2 * j)$  en zijn verdedigingskracht een willekeurige getal gelegen tussen 0 en  $(100 * i)$ .
  - Indien een samoerai nog  $i$  levens heeft en al  $j$  gevechten gewonnen heeft, is zijn aanvalskracht een willekeurig getal gelegen tussen 0 en  $(75 * i + 75 * j)$  en zijn verdedigingskracht een willekeurige getal gelegen tussen 0 en  $(80 * i + 80 * j)$ .

- Je kan nagaan of een krijger levend is m.a.w. of zijn aantal levens strikt groter is dan 0.
  - Het aantal levens waarover een krijger beschikt en zijn naam, worden meegegeven bij het aanmaken van het object.
  - Als een krijger een gevecht verliest, dan verliest hij één van zijn levens. Als hij wint, verhoogt het aantal gewonnen gevechten. Voorzie hiervoor methodes.
  - Twee krijgers kunnen ook vergeleken worden. De vergelijking gebeurt alfabetisch op basis van de naam en op basis van het al dan niet levend zijn. Krijgers met een zelfde naam kunnen maar gelijk zijn als ze allebei levend of allebei dood zijn. Is dat niet het geval dan komt de levende krijger eerst.
2. Schrijf een JUnit-test `KrijgerTest` die de toegevoegde functionaliteit uittest. Controleer alle mogelijkheden en de eisen geformuleerd in de Java API.
3. Schrijf een klasse `KrijgerVergelijker` die toelaat om twee krijgers te vergelijken. Het sorteren van krijgers met behulp van deze klasse gebruikt de onderstaande kenmerken (in volgorde):
1. Krijgers die levend zijn staan voor de krijgers die dood zijn.
  2. Krijgers met de meeste gewonnen gevechten staan vooraan.
  3. Krijgers met meer levens komen voor krijgers met minder levens.
  4. Krijgers waarbij alle bovenstaande kenmerken dezelfde zijn worden alfabetisch gesorteerd.

```
Michelangelo aantal gewonnen: 3 aantal levens: 4 levend
Arnold aantal gewonnen: 3 aantal levens: 3 levend
Bruce Lee aantal gewonnen: 3 aantal levens: 3 levend
Donatello aantal gewonnen: 2 aantal levens: 2 levend
Rafaelo aantal gewonnen: 2 aantal levens: 2 levend
Schreder aantal gewonnen: 3 aantal levens: 0 dood
```

4. Schrijf een JUnit-test `KrijgerVergelijkerTest` die de klasse `KrijgerVergelijkerTest` uittest. Je kan gebruikmaken van de volgende array's om krijgers aan te maken (zodat je de zes krijgers van hierboven bekomt). Plaats het aanmaken (en bewaren) van de krijgers in een aparte klasse `KrijgerDAODummy`.

---

```
final String[] namen = {"Donatello", "Michelangelo", "Rafaelo", "Schreder",
    "Bruce Lee", "Arnold"};
final int[] gewonnen = {2, 3, 2, 3, 3, 3};
final int[] levens = {2, 4, 2, 0, 3, 3};
```

---

5. Schrijf een abstracte klasse `KrijgersDAO` die een reeks krijgers bijhoudt. De initialisatie van de krijgers gebeurt in de afgeleide klasse. De klasse `KrijgersDAO` heeft drie methodes.
- Een methode die de krijgers teruggeeft.
  - Een methode die de krijgers teruggeeft gesorteerd volgens hun natuurlijke volgorde (alfabetisch en bij gelijke naam eerst de levende krijger). Het resultaat van deze methode is een `SortedSet<...>`. Kijk in de Java API hoe je deze klasse kan gebruiken.
  - Een methode die de krijgers teruggeeft gesorteerd volgens de volgorde bepaald in de klasse `KrijgerVergelijker` (zie vraag 3) Het resultaat van deze methode is een `SortedSet<...>`. Kijk in de Java API hoe je deze klasse kan gebruiken.

Gebruik de klasse `TreeSet<...>` als implementatie van de interface `SortedSet<...>`.

Pas `KrijgerDAODummy` aan zodat ze afleidt van de klasse `KrijgersDAO`.

6. Werk het hoofdprogramma `Krijgers` uit zodat het de krijgers drie keer uitschrijft: één keer in de volgorde dat ze aangeemaakt werden, één keer volgens hun natuurlijke volgorde en één keer volgens de volgorde beschreven in vraag 3. Maak hierbij gebruik van de klasse `KrijgersDAO`. Vermijd gedupliceerde code!

**Veel succes!**