



Noël Vaes

Java Trainer & Consultant



Nieuwigheden in Java 17 sinds Java 11

Roode Roosstraat 5
3500 Hasselt
België

+32 474 38 23 94
noel@noelvaes.eu
www.noelvaes.eu

Vrijwel alle namen van software- en hardwareproducten die in deze cursus worden genoemd, zijn tegelijkertijd ook handelsmerken en dienen dienovereenkomstig te worden behandeld.

Alle rechten voorbehouden. Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand of openbaar worden gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of op enige andere manier, zonder voorafgaande schriftelijke toestemming van de auteur. De enige uitzondering die hierop bestaat, is dat eventuele programma's en door de gebruiker te typen voorbeelden mogen worden ingevoerd opgeslagen en uitgevoerd op een computersysteem, zolang deze voor privé-doeleinden worden gebruikt, en niet bestemd zijn voor reproductie of publicatie.

Correspondentie inzake overnemen of reproductie kunt u richten aan:

Noël Vaes
Roode Roosstraat 5
3500 Hasselt
België

Tel: +32 474 38 23 94

noel@noelvaes.eu
www.noelvaes.eu

Ondanks alle aan de samenstelling van deze tekst bestede zorg, kan de auteur geen aansprakelijkheid aanvaarden voor eventuele schade die zou kunnen voortvloeien uit enige fout, die in deze uitgave zou kunnen voorkomen.

23/09/2021

Copyright© 2021 Noël Vaes



Inhoudsopgave

Hoofdstuk 1: Java versies.....	2
Hoofdstuk 2: Nieuwigheden in de Java-taal.....	4
2.1. Switch Expression.....	4
2.2. Tekstblokken.....	6
2.3. Records.....	8
2.3.1. Definitie van een <i>record</i>	9
2.3.2. Constructors.....	10
2.3.3. Methoden.....	12
2.3.4. Statische variabelen en methoden.....	13
2.3.5. Overerving.....	13
2.3.6. Records versus JavaBeans.....	13
2.4. <i>Pattern matching</i> met <i>instanceof</i>	14
2.5. Verzegelde (<i>sealed</i>) klassen en interfaces.....	17
Hoofdstuk 3: Nieuwigheden in het platform en tools.....	20
3.1. Packaging tool.....	20
3.2. Behulpzame <i>NullPointerException</i>	22
3.3. Interne API's ontoegankelijk.....	22
3.4. Floating Point bewerkingen.....	22
Hoofdstuk 4: Nieuwigheden in de Java-API's.....	23
4.1. Strings.....	23
4.2. <i>CompactNumberFormat</i>	23
4.3. Streams.....	24
4.4. Teeing Collector.....	24



Hoofdstuk 1: Java versies

Met de nieuwe *release*-strategie wordt er elk half jaar een nieuwe versie van Java uitgerold waarbij ook telkens het hoofdnummer wijzigt. Bijvoorbeeld Java 12, Java 13 enzovoort. Zo'n nieuwe versie bevat telkens drie soorten wijzigingen:

1. **Incubator modules:** De *incubator* is een soort proeftuin waar allerlei nieuwe dingen ontstaan en verder worden uitgewerkt. De bedoeling is om ontwikkelaars deel te laten uitmaken van dit proces. Ze kunnen dus naar believen spelen met deze dingen en er commentaar op geven om op deze manier de ontwikkeling vooruit te sturen. De API's die in die deze modules ontwikkeld worden zijn onaf en moeten verder uitgewerkt worden. Zulke *incubator modules* kunnen later ofwel geheel verdwijnen of evolueren naar een nieuwe *feature* in Java. *Incubator modules* worden gekenmerkt door de naam `jdk.incubator` in de modulenaam.
2. **Preview features:** Dit zijn nieuwigheden die inmiddels volledig uitgewerkt zijn, maar ze zijn nog niet definitief. Ze worden ter beschikking gesteld van de ontwikkelaars om ermee te experimenteren in de echte wereld en om na te gaan of er geen fundamentele problemen zijn bij het definitief toevoegen van deze *features* aan Java. Een *preview feature* kan dus mogelijk nog gewijzigd worden. Om gebruik te kunnen maken van deze *preview features* dient met de optie `--enable-preview` toe te voegen aan de *compiler* en de *jvm*.
3. **Permanent features:** Dit zijn definitieve en permanente toevoegingen aan Java. Deze *features* kunnen zonder problemen gebruikt worden in een productie-omgeving.

Om de drie jaar komt er een *Long Term Services Release* die een lange tijd verder ondersteund en onderhouden zal worden. Dit in tegenstelling tot de tussentijdse *releases* die slechts tijdelijk ondersteund en onderhouden zullen worden.

De laatste nieuwe LTS-*release* sinds Java 11 is Java 17 die uitgerold werd in september 2021.

Omwille van de langdurige ondersteuning van zo'n *release* is deze geschikt voor het gebruik in een productie-omgeving

In deze cursus bespreken we de geaccumuleerde *permanent features* in Java 17 sinds Java11. Dat zijn dus alle *permanent features* die toegevoegd werden in de *releases* 12, 13, 14, 15, 16 en 17.

De *incubator modules* en *preview features* laten we hier buiten beschouwing omdat ze niet geschikt zijn voor het gebruik in productie.

We maken tevens een selectie tussen de nieuwigheden die voor de doorsnee ontwikkelaar relevant zijn. Het is alleszins een exhaustief overzicht.

Deze nieuwigheden verdelen we verder onder in de volgende categorieën:

1. Nieuwigheden in de taal zelf.
2. Nieuwigheden in het platform en de *tools*.
3. Nieuwigheden in de API's.

We kunnen de JDK voor Java 17 downloaden van een van de volgende locaties:



- *Closed source* van Oracle: <https://java.oracle.com>.
- *Open source* van Oracle: <https://openjdk.java.net>.
- Andere *open-source-versies*: <https://adoptopenjdk.net>.

Enkele jaren geleden heeft Oracle de licentievoorwaarden voor de *closed-source*-versie gewijzigd zodat deze niet meer voor productiedoeleinden gebruikt kan worden zonder aangekochte licentie. Sinds Java 17 is deze versie opnieuw volledig vrij, ook voor het gebruik in productie.

Opdracht 1: JDK 17 installeren

- Haal de laatste versie van JDK 17 van de website: <https://java.oracle.com>.
- Installeer JDK 17 en accepteer hierbij de standaardinstellingen.
- Voeg een omgevingsvariabele toe met het pad naar de JDK.

Bijvoorbeeld op *Windows*: `JAVA_HOME=C:\Program Files\Java\jdk-17`

- Voeg vervolgens een pad toe aan de omgevingsvariabele `PATH` met een verwijzing naar de map `bin` van de JDK-installatie. Voeg dit pad bij voorkeur aan het begin toe zodat dit pad prioritair wordt.

Bijvoorbeeld: `PATH=%JAVA_HOME%\bin;...`

- Open een consolevenster en controleer de installatie met het volgende commando:

```
javac -version
```

- Configureer je IDE voor het gebruik van deze JDK. Dit is afhankelijk van de gebruikte IDE.



Hoofdstuk 2: Nieuwigheden in de Java-taal

2.1. Switch Expression

We kennen allemaal het *switch statement* waarbij we op basis van de waarde van een variable verschillende *statements* kunnen laten uitvoeren.

We illustreren het nog even met het volgende voorbeeld waarin we het aantal dagen berekenen voor een bepaalde maand in een bepaald jaar:

```
public static int getDaysInMonth(int month, int year) {
    int days = 0;
    switch (month) {
        case 2:
            days = (year%4==0)?29:28;
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            days = 30;
            break;
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            days = 31;
            break;
        default:
            days = 0;
    }
    return days;
}
```

Dit *switch statement* wordt gekenmerkt door het *fall-through*-scenario waarbij bij een bepaalde *case* de daaropvolgende code wordt uitgevoerd tot het eerstvolgende *break statement* of het einde van het *switch statement*.

De code is evenwel behoorlijk omslachtig en in dit geval kunnen we gebruikmaken van de nieuwe *switch expression*.

We brengen even in herinnering dat een *expression* (uitdrukking) een constructie is die enkel een waarde genereert die dan vervolgens gebruikt kan worden in een andere *expressions* of een *statement*. Zo zal de *switch expression* ook gewoon een bepaalde waarde teruggeven op basis van een variabele die bepaalde waarden (*cases*) kan aannemen. De werking is daarom ook anders dan een *statement* dat een volledige programmeerregel bevat.

De algemene syntax is als volgt: