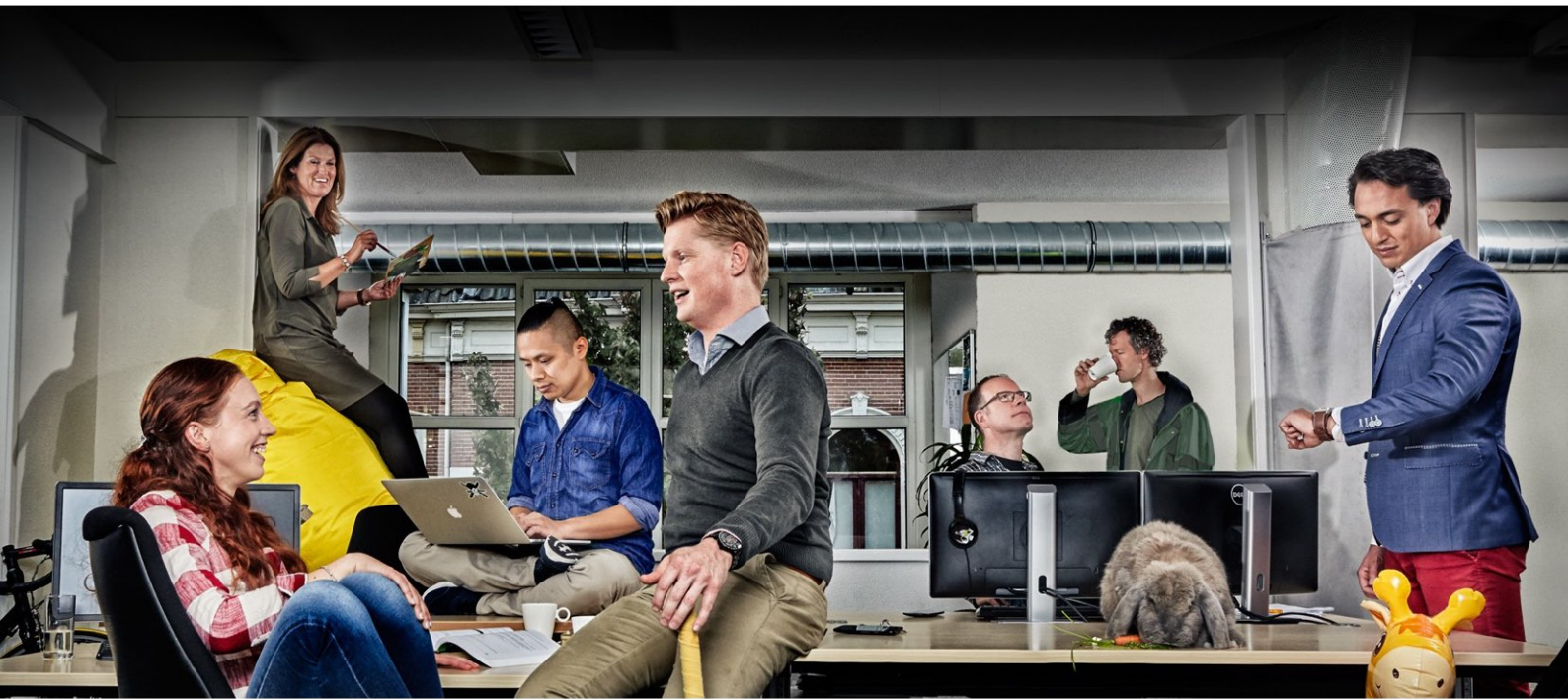


topicus
ACADEMY



TA-101

Introductie Ontwikkelaar

Docentenhandleiding



TA101 Introductie Ontwikkelaar

Topicus Academy

Inhoudsopgave

Cursusomschrijving	1
Inleiding	1
Inhoud	1
1. Introductie software engineering	2
1.1. Inleiding	2
1.2. Voorbereiding	2
1.3. Inhoud	2
2. Leer de commandline gebruiken	5
2.1. Inleiding	5
2.2. Voorbereiding	5
2.3. Inhoud	5

Cursusomschrijving

Inleiding

De bootcamp voor ontwikkelaars begint met een uitleg over hoe een ontwikkelaar zijn werk doet. Er komen nogal wat begrippen langs tijdens het ontwikkelen, en deze sessie heeft als doel om die begrippen te duiden en een plek te geven.

Inhoud

In deze module behandel je de begrippen die veel voorkomen bij het ontwikkelen. Denk hierbij aan de begrippen die in de onderstaande tabel staan. Deze tabel is niet leidend, bereid het gerust uit of laat er een aantal weg. Voor de uitleg maak je gebruik van een whiteboard, waarop je alles uittekent. Bespreek de begrippen en leg de onderlinge samenwerking uit (vergeet niet de verbindende lijnen te tekenen).

Vraag veel aan de trainee(s) en laat ze veel vragen stellen. Neem hier gerust 1-3 uur voor.

Neem een foto van het whiteboard (eventueel op gezette tijden) om het resultaat op te slaan.

De termen die voorbij komen wanneer je een Java ontwikkelaar wordt zijn onder andere de volgende:

Compiler	Database	EAR	Git
github	Hibernate	JAR	Java
Java EE	JDBC	Jenkins	JPA
Maven	OTAP	SQL	Tomcat
WAR	Wicket	Wildfly	

Chapter 1. Introductie software engineering

1.1. Inleiding

In deze module vertel je bij het whiteboard hoe je van Java broncode naar een werkend programma komt (afhankelijk van de voorkennis van de trainee(s)).

1.1.1. Doelen

- Hoe schrijf je Java code
- Wat is Java en de Java Virtual Machine
- Hoe deel je je code met andere team leden
- Hoe wordt jouw Java code een programma
- Hoe zit een programma als ParnasSys, SOMtoday of EduArte in elkaar
- Hoe wordt een systeem als ParnasSys gebouwd

1.1.2. Voorkennis

- Geen

1.2. Voorbereiding

- Zorg voor een whiteboard
- Stel de student van te voren een aantal vragen om er achter te komen wat het niveau is

1.3. Inhoud

1.3.1. Van code naar applicatie

Leg hierbij uit wat Javac is en doet, wat een JVM is en doet en wat een JAR file is en doet. Teken het proces van source file naar class file en de rol van Eclipse.

Teken een applicatie met een database aan de onderkant, en een browser aan de bovenkant.

Leg uit wat de database doet en wat SQL daarbij voor rol speelt. Bijv. tabellen en kolommen. Leg uit dat het programma werkt met begrippen als Leerling en dat een Leerling gedrag heeft. Leg uit dat een tabel kolommen heeft als geboortedatum en naam, en dat je om de gegevens van de database naar de applicatie en de gebruiker te krijgen je gebruik maakt van code van anderen (Hibernate) en dat dat aangeleverd wordt als een JAR bestand, dat we gebruiken in onze applicatie.

Leg uit dat de applicatie o.a. HTML uitspuugt en daarvoor gebruiken we Wicket om van Java naar HTML te gaan, ook een bibliotheek van externen.

Om deze bibliotheken te beheren gebruiken we Maven. De bibliotheken komen van een centrale plek op het internet, en wij hebben daar een kopie van onder de naam Artifactory. Teken dit uit op

het bord.

Reden voor Artifactory: als iedereen op de wereld met de centrale plek zou communiceren om de gegevens op te halen kost dat erg veel geld, en bandbreedte, dus maken we voor onze 50 interne ontwikkelaars een lokale kopie, die zelf met één connectie naar de centrale plek werkt.

Leg ook uit dat je vanuit huis niet zomaar bij onze Artifactory komt omdat daar ook onze eigen broncode op staat, en dat we niet willen dat Magister daar zomaar bij kan. Daar heb je het VPN voor nodig.

1.3.2. Applicatie architectuur

Leg uit wat een applicatie bij ons doet: JSON, JPGs, PDFs, HTML uitspuwen, JavaScript en CSS.

Daarna leg je uit hoe een applicatie er gelaagd uit ziet. Sta erbij stil dat de bovenste laag alleen HTML genereert voor de browser, of berichten voor andere applicaties (XML, JSON, iOS applicaties, of bijv. DUO). Leg uit dat er een database onder de applicatie ligt, en dat er soms een scheiding is tussen de database en REST aan de ene kant, en de HTML generende applicatie (of iOS/Android applicaties) aan de andere kant.

Leg uit dat de database Oracle, MySQL, PostgreSQL of SQLServer kan zijn. Leg uit dat voor de communicatie gebruik gemaakt wordt van JDBC. Hibernate gebruikt JDBC om van SQL Java objecten te maken.

Geef aan dat er verschillende lagen in de applicatie zijn:

- Data laag
- Business laag
- UI laag

Geef aan dat het Java proces op de server "Applicatie server" heet, en bij ons door Tomcat of Wildfly geïmplementeerd is. Illustreer de rol van de server door aan te geven dat deze honderden, duizenden gebruikers van de applicatie in goede banen leidt en er voor zorgt dat er op een standaard manier gecommuniceerd kan worden met andere componenten.

1.3.3. Ontwikkelproces

GIT

Naast het schrijven van code moet je je code wel delen met anderen. Daarvoor gebruiken we Git. Leg uit hoe git hier in dit proces werkt met github en je lokale laptop.

Vergeet ook niet te zeggen dat wanneer het alleen op je laptop staat, de code niet bestaat (want bus → geen code, koffie over laptop → geen code). Code moet naar github.

Hoe werk je ongeveer per dag? Haal wijzigingen van team binnen naar lokaal, doe je eigen wijziging, controleer, deel wijziging via pull request, code review, pas aan, deel, code review, repeat, test en naar master en de rest van het team.

Jenkins

Leg uit dat we een buildserver hebben met de naam Jenkins, en dat deze elke wijziging opnieuw bouwt om te controleren of er niets mis is.

OTAP

Dan gaat de code van ontwikkel naar test, naar acceptatie naar productie. Leg dit proces uit.

SCRUM

Leg kort het proces van SCRUM uit. Planning, sprint, daily standups, iteraties, demo en retrospective.

JIRA voor de bugs

Meldingen van gebruikers komen in JIRA, als je branches en dergelijke goed noemt, worden die automatisch gekoppeld aan JIRA zodat je dat later weer kan terug zien.

Chapter 2. Leer de commandline gebruiken

2.1. Inleiding

In deze module leert de student doormiddel van zelfstudie werken met de commandline. Dat betekent NIET dat je als docent achteruit kunt gaan zitten. Deze module vraagt zeker wat voorbereiding voor de docent.

2.1.1. Doelen

- De basis commando's kennen voor de commandline
- Nieuwe folders te maken
- Door deze folders te navigeren
- Informatie te vinden
- Bestanden te kopieëren of te verwijderen
- Bestanden in te zien.

2.1.2. Voorkennis

- Geen

2.2. Voorbereiding

- Zorg ervoor dat je weet waar de student mee aan het werk gaat, zodat je snel en doelgericht mogelijke vragen kan beantwoorden.
- Check van te voren of alle links in de leerlinghandleiding nog up-to-date zijn en lees ze indien nodig zelf even door.

2.3. Inhoud

De leerlingen heeft de volgende bronnen opgekregen om door te nemen en mee aan de slag te gaan:

- Lees <http://www.chronicle.com/blogs/profhacker/getting-comfortable-on-the-command-line/36862>
- [] Zed Shaw. Learn Python the Hard Way. <https://learnpythonthehardway.org/book/appendixa.html> 2016
- <http://www.chronicle.com/blogs/profhacker/start-doing-things-on-the-command-line-with-script-ahoy/62489>