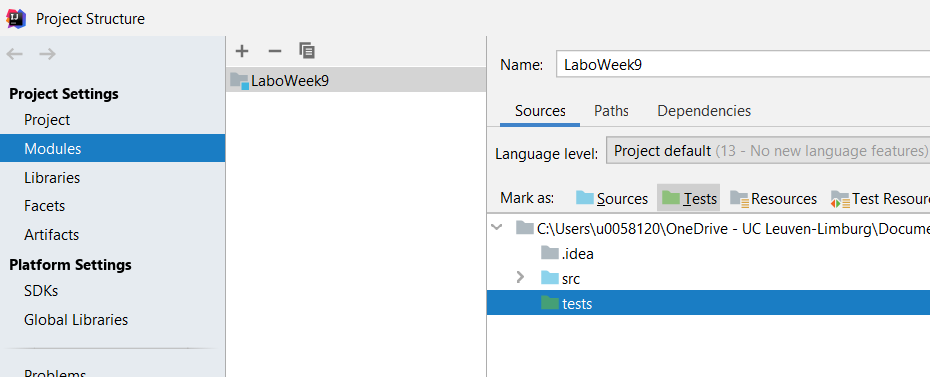
# Labo bestanden en List

In dit labo leer je om te werken met bestanden en de List-interface

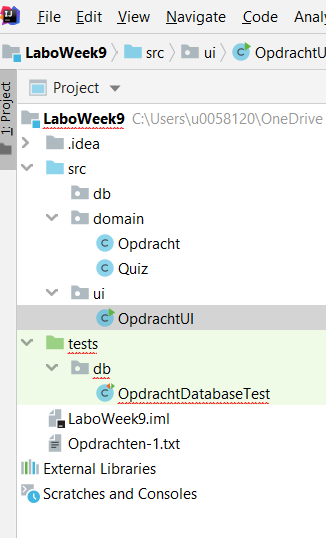
## Voorbereiding

* Maak een nieuw project in IntelliJ met de naam LaboWeek9.
* Maak in de src map 3 packages met de namen *domain*, *ui* en *db*.
* Maak een nieuwe directory in je project met de naam *tests*.
* Duid via File 🡪 Project Structure 🡪 modules aan dat deze nieuwe directory de test-klassen zal bevatten.



* Maak in de directory tests een package met de naam *db* en kopieer hierin het gegeven bestand OpdrachtDataseTest.java. Ga in deze klasse op @Test staan 🡪 druk Alt+enter en voeg JUnit4 toe.
* Voeg de gegeven (onvolledige) bestanden Opdracht.java en Quiz.java toe aan de domain-package van de src-map.
* Voeg het gegeven bestand OpdrachtUI.java toe aan de ui-package van je src-map.
* Ga op je projectnaam staan in IntelliJ en klik. Kopieer het gegeven txt bestand opdrachten-1.txt en paste het in je project.

De structuur van je project moet er nu zo uitzien.



## Deel 1

We maken een kleine applicatie voor een quiz. Bekijk ter inspiratie de vragen die je in het bestand opdrachten-1.txt vindt.  
  
Om te starten krijgen jullie een klasse Opdracht, waar een mini ui (OpdrachtUI) werd rond geschreven.

Daarnaast krijgen jullie een klasse Quiz. Een quiz bevat een collectie van opdrachten. De code voor deze klasse is echter onvolledig.

* Vul de klasse Quiz aan.
* Merk op dat de methode *equals()* in de klasse Opdracht niet volledig is. Vervolledig ze.

Jullie krijgen ook klasse OpdrachtDatabankTest. Deze klasse compileert niet aangezien de OpdrachtDatabank nog niet bestaat.

* Schrijf de klasse OpdrachtDatabank met hierin de code om alle opdrachten in Opdrachten-1.txt in te lezen en terug tegeven. Zorg dat de testen slagen.
* Hint: de verschillende onderdelen in het .txt bestand worden onderling gescheiden door een Tab; gebruik hiervoor “[\\t](file:///\\t)” als regex bij de split-method op de invoerstring dus s.split(“[\\t](file:///\\t)”)

Demonstreer je code door de main methode verder uit te breiden als volgt:

* Laat alle ingelezen opdrachten, met hun juiste antwoord zien op de console
* Kies een willekeurige opdracht en toon de vraag aan de gebruiker in de console, waarbij hij een antwoord kan geven dat gecontroleerd wordt met het juiste antwoord.

## Deel 2

De databank klasse moet de opdrachten ook kunnen geven gesorteerd op basis van categorie en voor eenzelfde categorie op basis van opdracht\_ID.

* Vul de klasse OpdrachtDatabankTest aan. Voeg een testmethode toe die een (nieuw te maken) methode *getOpdrachtenGesorteerdOpCategorie()* test. Voeg ook een testmethode toe die een (nieuw te maken) methode *getOpdrachtenGesorteerdOpCategorieEnId()* test.
* Vul de klasse OpdrachtDatabank aan. Voeg de methoden *getOpdrachtenGesorteerdOpCategorie()* en *getOpdrachtenGesorteerdOpCategorieEnId()* toe.