

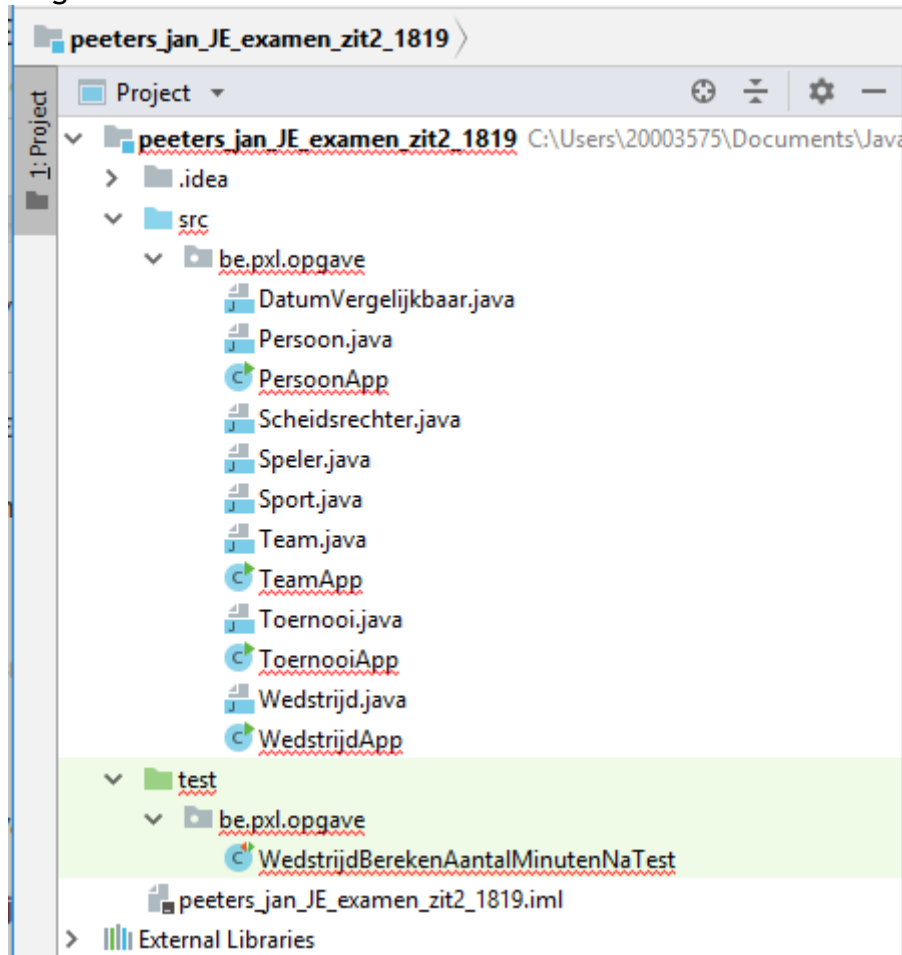
STAPPEN UIT TE VOEREN VOOR JE AAN HET LAPTOPGEDEELTE BEGINT

Maak in IntelliJ een project met als naam `<familienaam_voornaam_JE_examen_zit2_1819>`.

Voorbeeld: een student met naam Jan Peeters maakt een module met de naam **`peeters_jan_JE_examen_zit2_1819`**

Maak naast de src directory ook een test directory aan. Maak in beide directories een package aan met naam: `be.pxl.opgave`

De startbestanden download je van FileZilla en je voegt ze toe aan je project zodat je de volgende startsituatie bekomt:



Zodra je de signatuur van een klasse aanmaakt, zal de extensie `.java` verdwijnen. Plaats bovenaan in elke file je naam in commentaar.

Opgave (75 punten)

1001 0110	5%
--------------	----

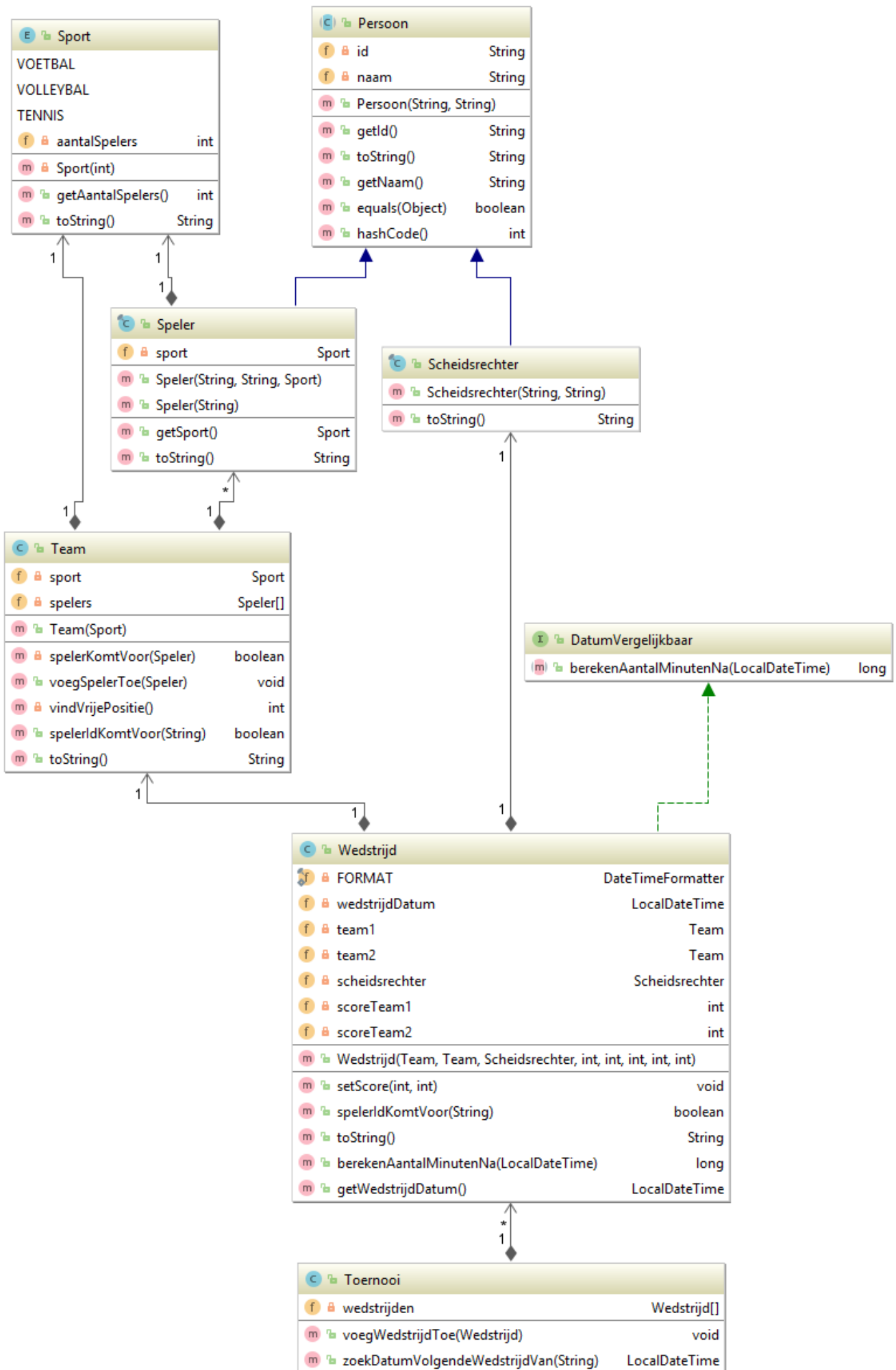
{ CC }	2% max 4%
--------	--------------

✓ SPEC	2% max 4%
--------	--------------

Schrijf de nodige code op basis van het gegeven klassendiagram en hou rekening met opmerkingen op volgende pagina's. Je mag GEEN extra eigenschappen, constructoren of andere methoden voorzien tenzij het expliciet vermeld wordt in de opgave.

Hergebruik code waar mogelijk, vermijd het gebruik van dubbele code!!!

Het klassendiagram vind je ook als afbeelding in het mapje dat je van FileZilla downloadt.



Sport

Binnen de enum Sport wordt de eigenschap aantalSpelers bijgehouden. De mogelijke objecten van Sport zijn VOETBAL, VOLLEYBAL en TENNIS. VOETBAL heeft 11 spelers, VOLLEYBAL 8 en TENNIS 2.

Via de constructor krijgt de eigenschap aantalSpelers een waarde.

Er is ook een getter voorzien voor aantalSpelers.

Voorzie ook de toString-methode. Deze methode geeft de eerste 3 letters van de naam van het object in kleine letters terug.

Persoon

Er kunnen geen objecten van de klasse Persoon gemaakt worden.

Persoon heeft eigenschappen id (String) en naam (String).

Voorzie een constructor waarin de waarde voor id en naam als argument meegegeven wordt. Uit het ingegeven id worden alle symbolen, die niet een letter of een getal zijn, verwijderd. Het id mag ook niet langer dan 3 symbolen zijn. Indien dit wel zo is, worden enkel de eerste 3 symbolen van opgegeven id aan de eigenschap id toegekend. Verder worden alle letters in de opgegeven waarde voor id omgezet naar hoofdletters.

Enkele voorbeelden van ingegeven en toegekende waarden vind je terug in de onderstaande tabel.

Ingegeven id	Waarde van de eigenschap id
3@t\$ABC"	3TA
a@9	A9
1 @Abxyz	1AB

Maak getters voor id en naam.

Zorg ervoor dat de Persoon vergeleken kan worden met eender welk object zodanig dat de persoon gelijk beschouwd wordt aan het object als het object een Persoon is en de id van het object gelijk is aan de id van de persoon.

In de methode toString wordt een String representatie van de vorm

[A12] Tim Schuermans

teruggeven voor een Persoon met naam Tim Schuermans en id A12.

Speler

Speler is afgeleid van de klasse Persoon. Van deze klasse kan geen nieuwe klasse afgeleid worden.

Spelers hebben eigenschappen id (String), naam (String) en sport (Sport).

Maak een eerste constructor waarbij elke eigenschap zijn waarde krijgt.

In de tweede constructor wordt enkel een waarde voor id ingegeven, naam krijgt een lege String als waarde, sport krijgt null als waarde. Roep in deze constructor de eerste constructor aan.

Maak een getter voor Sport.

Voorzie de toString-methode. De teruggekeerde String is van de vorm

[B] Sofie Vanhentrijk (ten)

voor een Speler met id B, naam Sofie Vanhentrijk en sport TENNIS.

Scheidsrechter

Scheidsrechter is afgeleid van de klasse Persoon. Deze klasse kan niet overgeërfd worden.

De klasse scheidsrechter heeft eigenschappen id (String), naam (String).

Maak een constructor waarbij elke eigenschap zijn waarde krijgt.

Voorzie de toString-methode. De teruggekeerde String is van de vorm

[3Z2] Dirk Smeyers <ref>

voor een Scheidsrechter met id 3Z2 en naam Dirk Smeyers.

Team

De klasse Team heeft sport (Sport) en spelers (array van Speler-objecten) als eigenschappen.

In de constructor wordt een waarde voor sport ingegeven. De array spelers wordt ook aangemaakt. Er worden evenveel plaatsen in deze array voorzien als de waarde voor aantalSpelers van de eigenschap sport.

Via de methode voegSpelerToe wordt een speler toegevoegd aan de array spelers op voorwaarde dat

- de speler dezelfde sport heeft als de sport van het team
- er nog geen speler met hetzelfde id al toegevoegd is aan de array spelers
- er nog ruimte vrij is in de array spelers (er een null waarde in de array voorkomt)

Als er niet aan deze voorwaarden voldaan is, wordt dit gemeld (afgedrukt in de console) als “Foutieve ingave”.

Maak in de uitwerking van voegSpelerToe gebruik van de private methoden spelerKomtVoor en vindVrijePositie.

De methode spelerKomtVoor geeft true of false terug naargelang de speler die meegegeven wordt als argument al dan niet in de rij spelers staat.

De methode vindVrijePositie geeft een int terug met daarin de index van de eerste null-waarde in de eigenschap spelers. Als er geen null in de array staat, dan wordt de waarde -1 teruggegeven.

Daarnaast voorzie je de methode spelerIdKomtVoor. Deze wordt gebruikt om te controleren of er een speler met het opgegeven id in het team aanwezig is. Deze methode heeft spelerId (String) als parameter. Er wordt een speler aangemaakt met dit id en via de equals-methode van de klasse Speler wordt gecontroleerd of deze speler in het team aanwezig is.

In de toString-methode wordt een String teruggegeven van de onderstaande vorm:

Onvolledig team

wanneer de array spelers nog niet volledig opgevuld is of

[B] Sofie Vanhentrijk (ten)

[QA2] Tine Verhulst (ten)

wanneer de array wel volledig opgevuld is (in dit voorbeeld met 2 tennisspelers).

DatumVergelijkbaar

Maak de interface DatumVergelijkbaar. Binnen de interface wordt de methode berekenAantalMinutenNa aangeboden. Deze methode heeft een LocalDateTime als argument en geeft een long terug.

Wedstrijd

De klasse Wedstrijd implementeert de interface DatumVergelijkbaar.

De eigenschappen van Wedstrijd zijn:

- wedstrijdDatum (LocalDateTime)
- team1 (Team)
- team2 (Team)
- scheidrechter (Scheidsrechter)
- scoreTeam1 (int)
- scoreTeam2 (int)

In de constructor wordt een waarde voorzien voor team1 (Team), team2 (Team), scheidsrechter (Scheidsrechter), dag (int), maand (int), jaar (int), uur (int) en minuut (int). Aan de hand van dag, maand, jaar, uur, minuut wordt een LocalDateTime aangemaakt en toegekend aan wedstrijdDatum.

Via de methode spelerIdKomtVoor, die een spelerId (String) als argument heeft, wordt gekeken of een speler met opgegeven spelerId al in team1 of team2 aanwezig is. Indien dit zo is, wordt true teruggegeven, anders false. Hou er rekening mee dat team1 en team2 gelijk kunnen zijn aan null.

Aan de hand van de methode berekenAantalMinutenNa wordt het aantal minuten tussen de wedstrijddatum en een LocalDateTime die als argument meegegeven wordt, berekend en teruggegeven als long. Indien de meegegeven datum vóór de wedstrijddatum komt, krijg je een negatief getal. Indien de meegegeven datum na de wedstrijddatum komt, krijg je een positief getal. In het startbestand vind je unit testen om deze methode te testen.

Voorzie de getter voor wedstrijdDatum. Voorzie eveneens een setter om de score van de wedstrijd in te vullen.

In de toString methode wordt onderstaande weergave gegenereerd:

```
TEAM1
[A12] Tim Schuermans (ten)
[B23] Mieke Verlaek (ten)
TEAM2
[FDS] Hilde Voorts (ten)
[QEE] Marleen Elders (ten)
01/07/19 09:05
SCHEIDSRECHTER [AQD] Jef Schepis <ref>
SCORE 4 3
```

De score mag je enkel tonen als de wedstrijd reeds gestart of gespeeld is. Indien de wedstrijd nog niet begonnen is, druk je de score niet af.

Toernooi

De klasse Toernooi bevat een array van Wedstrijd-objecten. Verder bevat de klasse ook een constante MAX_AANTAL_WEDSTRIJDEN met waarde 10.

De constructor heeft geen argumenten. In de constructor wordt de rij geïnitieerd zodanig dat hij MAX_AANTAL_WEDSTRIJDEN wedstrijd-objecten kan bevatten.

Via de methode `voegWedstrijdToe` wordt een `Wedstrijd`-object toegevoegd aan de array `wedstrijden` op voorwaarde dat er nog ruimte vrij is in de array `wedstrijden` (er een null waarde in de array voorkomt). Als er geen ruimte meer is in de array `wedstrijden` wordt de foutmelding 'geen ruimte meer voor wedstrijden' afgedrukt.

Via de methode `zoekDatumVolgendeWedstrijdVan` wordt een `LocalDateTime` teruggegeven met daarin de datum van de volgende wedstrijd van de speler wiens `spelerId` (`String`) opgegeven is als argument. Als er geen wedstrijd gevonden wordt, wordt null teruggegeven.