# **Naschoolse Activiteiten**

Deze herhalingsoefening herhaalt hoofdzakelijk de leerstof van voor de paasvakantie. Ze geldt tevens als voordbeeldexamen voor deze leerstof. De opzet en instructies zijn gelijkend op hetgeen je bij één deel van het examen zal krijgen.

**Opzet van je project**

* Maak een nieuw project aan. Noem het naamVoornaam.
* Download de zip met gegeven bestanden. Unzip.
* Maak in de src map van je project een package met de naam domain
* Kopieer de bestanden in de domain package
* Maak in de src/domain map een klasse DomainException die opgegooid wordt overal waar zich in de klassen van de domain-package een exception voordoet.
* Vul de code in de src/domain-klassen aan of verbeter deze overeenkomstig het verhaal op volgende pagina’s. Denk hierbij aan
  + final, static, abstract, protected daar waar dit kan/moet
  + voorzie de nodige interface(s)
  + gooi je zelf gemaakte DomainException daar waar nodig
* Maak een testklasse voor de klasse OrganisatieNaschoolseActiviteiten waarin je alle testcases voor de functionaliteit schrijfIn() implementeert.

Een school heeft jou gevraagd om een applicatie te ontwerpen om informatie over hun **naschoolse activiteiten** te beheren. De school wenst een lijst bij te houden van alle georganiseerde activiteiten. Hiervoor is het nodig om activiteiten aan de lijst te kunnen toevoegen. Een gegeven activiteit kan maar één keer toegevoegd worden aan de lijst. Een activiteit heeft een naam (niet wijzigbaar) en gaat door op een bepaalde datum (LocalDate). Naam en datum mogen niet leeg zijn. Beide zijn opvraagbaar. Momenteel zijn de volgende soorten activiteiten mogelijk: dropping, wedstrijd en speelplein.

Er zijn activiteiten waarvoor uit praktische overwegingen moet ingeschreven worden om te kunnen deelnemen. Voor elke activiteit waarvoor ingeschreven moet worden, moeten volgende functionaliteiten mogelijk zijn:

* Nagaan of een gegeven persoon is ingeschreven
* Inschrijven van een persoon.
  + Een persoon kan zich maar één keer inschrijven voor een gegeven activiteit
  + Er zijn activiteiten waarvoor een minimumleeftijd vereist is. Inschrijven voor zo een activiteit kan enkel indien de persoon de vereiste leeftijd heeft op de dag dat de activiteit doorgaat.
  + Deze methode gooit een zelf te schrijven DomainException als de inschrijving voor deze activiteit niet correct verliep.

Van een **persoon** worden naam, voornaam en geboortedatum bijgehouden. De gegevens van een persoon mogen niet wijzigbaar zijn. Je moet de leeftijd van een persoon kunnen raadplegen op een gegeven dag.

Een **dropping** heeft een startplaats (string). Om deel te nemen aan een dropping dient een persoon zich in te schrijven. Een inschrijving voor een dropping kan enkel doorgaan indien de gegeven persoon minstens de vereiste minimumleeftijd heeft. Voor elke dropping geldt de minimumleeftijd van 10 jaar.

Een **wedstrijd** activiteit heeft een bijhorende sport. Ook voor een wedstrijd activiteit dienen personen zich in te schrijven. Voor het inschrijven voor een wedstrijd activiteit geldt enkel een minimumleeftijd indien de bijhorende sport dit oplegt. Een inschrijving voor een wedstrijd kan dus enkel doorgaan indien de gegeven persoon minstens de minimumleeftijd heeft die wordt opgelegd door de bijhorende sport.

Een **sport** heeft een naam die niet leeg mag zijn. Op dit moment zijn er 2 mogelijke sporten: Voetbal en Lopen. Voor **Voetbal** geldt altijd een minimumleeftijd van 5 jaar. Voor **Lopen** geldt geen minimumleeftijd. Uiteraard zullen er in de toekomst nog andere mogelijke sporten bijkomen, eventueel met minimumleeftijd.

Voor een **speelplein-activiteit** geldt geen minimumleeftijd en dient niet ingeschreven te worden.

Van elke activiteit moeten alle eigenschappen uitgeschreven kunnen worden (zie bijgevoegd voorbeeld).

Voor elke activiteit moet gecontroleerd kunnen worden ze beschikbaar is voor een gegeven persoon.

Uiteraard zullen er in de toekomst nog andere mogelijke activiteiten bijkomen, eventueel met een inschrijvingslijst, eventueel met een minimumleeftijd.

We wensen voor de naschoolse activiteiten organisatie volgende functionaliteiten:

* Het aantal geregistreerde activiteiten opvragen
* Een nieuwe gegeven activiteit toevoegen aan de lijst met activiteiten.
* Een gegeven persoon inschrijven voor een gegeven activiteit. Indien de gegeven activiteit niet voorkomt in de kalender moet een foutboodschap gegenereerd worden. Als de gegeven persoon niet aan de nodige voorwaarden voldoet, gooit de methode de foutboodschap van de klasse Activiteit door.
* Een lijst teruggeven met alle wedstrijd-activiteiten;
* Een collectie geven van mogelijke activiteiten voor een gegeven persoon op een gegeven dag, rekening houdend met zijn leeftijd
* Alle activiteiten van een gegeven dag verwijderen. Deze methode geeft de lijst met verwijderde activiteiten terug.

**Extra**

De opdracht hieronder maakt **geen deel van het examen**, maar is wel nuttig om je code te testen.

Je krijgt van ons een main methode die objecten aanmaakt met hardgecodeerde gegevens. Pas de methode aan zodat opgegooide fouten naar de console geschreven worden, maar het programma nadien verder loopt.