



<p>Naam student: _____</p> <p>Klas: 5 ICW</p> <p>Schooljaar: 2024 - 2025</p> <p><input type="checkbox"/> REDICODIS</p>	<p>Datum: 12 juni 2025</p> <p>Evaluatie:     <b>/ 130</b></p>
--	---

Vraag 1: In bijlage 1 vind je een overzicht van 8 verzamelkaarten (Pokémon en Yu-Gi-Oh!). Afbeeldingen zijn ook groter te bekijken op Smartschool). De bedoeling is dat je een applicatie maakt die deze spelkaarten kan bijhouden.

       / 30

Maak één of meerdere gepaste klasse(s) aan om deze kaarten in voor te stellen. Je gaat enkel af van de data die op de geleverde kaarten staan, enige andere, persoonlijke voorkennis over deze verzamelkaarten is hier niet van toepassing. Je probeert zo bedachtzaam mogelijk om te gaan met aantal properties en deze te hergebruiken waar mogelijk.

/ 15

Je zorgt ervoor dat je niet zomaar een verzamelkaart kan aanmaken, maar enkel concrete kaarten. Dit geldt ook voor enige andere klassen die je gebruikt maar die niet alle data van een bepaalde kaart voorstellen.

/ 3

**TIP: kopieer/plak zoveel mogelijk gemeenschappelijke code voor de constructoren van klassen met dezelfde superklasse.**

Je stelt één (1) structuur op zodat je per kaart het aantal stuks hiervan kan bijhouden. Je vult deze op met de 8 kaarten in bijlage d.m.v. van het oproepen van bijhorende constructoren. Bij lange teksten mag je enkel de eerste 3 woorden opschrijven, gevolgd door ... (bv. bij Pikachu: "Search your deck...")

/ 6

Je toont alle Pokémonkaarten met het aantal daarvan en vervolgens alle Yu-Gi-Oh!-monsters (let op: niet afzonderlijke kaarten) met hierbij ook het aantal.

/ 6

Pokémonkaarten met aantal:  
Pikachu (MEW EN 025/165) - 1  
Metang (Ster 94/168) - 1  
Water (XY 93/108) - 1  
Gym Trainer ([ ] 059/72) - 1

Yu-Gi-Oh monsterkaarten:  
Blue-eyes white dragon - ATK / 3000 DEF / 2500: 2  
Kuriboh - ATK / 300 DEF / 200: 1



Vraag 2:

/ 30

Je hebt een klasse **Ontwikkelaar** die naast een *naam* een *lijst* van **Game-objecten** bevat.

Elke **Game** heeft de volgende eigenschappen: titel, genre, duur (gemiddelde speelduur), releasedatum, platform, beoordeling, franchise en ondersteunde talen. Daarnaast heeft een game een lijst van **Personage**-objecten en een lijst van **Achievement**-objecten.

Een **Personage** heeft de volgende eigenschappen: naam, rol, omschrijving, leeftijd, speciale vaardigheid en een indicatie die aangeeft of het speelbaar is.

Een **Achievement** bevat een naam, omschrijving, aantal sterren en een indicator of het verborgen is.

In het Program.cs-bestand vind je een lijst van ontwikkelaars. Haal d.m.v. LINQ-functies hieruit de volgende queries, op een zo'n performant mogelijke manier. Je mag eerdere queries hergebruiken (voortbouwend) in volgende queries. Het resultaat van de query moet steeds in een variabele zitten.

*Als je voor de eerste 3 queries het antwoord niet kent, mag je dit vragen aan de toezichter. Deze queries worden dan niet beoordeeld (0).*

- Lijst van alle games ( /1)
- Lijst van alle personages ( /1)
- Lijst van alle achievements ( /1)
- Naam van ontwikkelaar met de minste games (er is er maar 1) ( /1)
- Naam van de oudste game ( /3)
- Games uitgebracht tussen 2010-2019, gesorteerd op releasedatum ( /3)
- Verborgen achievements ( /1)
- Speelbare personages uit de franchise 'The Last of Us', gesorteerd op naam ( /5)
- De moeilijkste achievements uit de 'Super Mario'-franchise ( /3)
- Speelbare personages gesorteerd op leeftijd (oud naar jong), dan alfabetisch op naam, waarbij de naam wordt getoond en tussen haakjes de leeftijd ( /5)
- Hebben alle games een achievement van de gemakkelijkste graad (1)? ( /2)
- Boxplot van de duurtijden van de games ( /4)
- BONUSVRAAG: alle games per genre met een rating van minstens 8 ( /3)

Aantal games: 12

Aantal personages: 32



Aantal achievements: 24

Ontwikkelaar met minste games: CD Projekt Red

Oudste game: Half-Life 2

Games tussen 2010-2019, gesorteerd op releasedatum:

Portal 2: 19/04/2011

The Elder Scrolls V: Skyrim: 11/11/2011

Grand Theft Auto V: 17/09/2013

The Witcher 3: Wild Hunt: 19/05/2015

Uncharted 4: A Thief's End: 10/05/2016

The Legend of Zelda: Breath of the Wild: 3/03/2017

Super Mario Odyssey: 27/10/2017

Red Dead Redemption 2: 26/10/2018

Verboren achievements:

Criminal Mastermind (5)

Speelbare personages uit The Last of Us-franchise, gesorteerd op naam:

Ellie (Held)

Joel (Held)

Moeilijkste achievements uit de Super Mario-franchise:

Power Moon Collector (4)

Power Star Collector (4)

Speelbare personages gesorteerd volgens leeftijd (oud naar jong), dan naam (alfabetisch):

Geralt of Rivia (100)

Joel (48)

Michael De Santa (45)

Mario (40)

Trevor Philips (40)

Nathan Drake (38)

Arthur Morgan (36)

Dragonborn (30)

Chell (27)

Gordon Freeman (27)

Franklin Clinton (25)

Link (17)

Ellie (14)

Hebben alle games een achievements van de gemakkelijkste graad? Nee

minimum|--|Q1--mediaan--Q3|--|maximum

10:00:00|--|15:00:00--1.01:00:00--2.12:00:00|--4.04:00:00

Games per genre met een beoordeling van minstens 8:

Action: Grand Theft Auto V

Action-Adventure: Red Dead Redemption 2, The Last of Us Part I, The Last of Us Part II

Adventure: The Legend of Zelda: Breath of the Wild

Platformer: Super Mario Galaxy, Super Mario Odyssey

RPG: The Elder Scrolls V: Skyrim, The Witcher 3: Wild Hunt



GO! SCHOOL VOOR BUITENGEWOON  
SECUNDAIR ONDERWIJS OPLEIDINGSVORM 4 TYPE 9

Campus Impuls

**Examen**  
**Informaticawetenschappen**  
**Software**

Vraag 3:

Maak een klasse 'Persoon' met 3 eigenschappen: voornaam, achternaam en geboortedatum (DateTime). Deze klasse geef je een functie die de leeftijd (aantal jaar) van een persoon berekend.

Je laat vervolgens de gebruiker een persoon aanmaken. Hierbij vraag je de voornaam, achternaam, geboortedag, geboortemaand en geboortjaar. Tenslotte toon je de naam van de gebruiker, alsook diens leeftijd.

Je vangt eventuele fouten op en toont een gepaste melding.

/ 15

/ 6

/ 2

/ 7



Vraag 4:

Maak een console applicatie die een connectie maakt met een MySQL-databank. Deze applicatie dient voor een hotel, waarbij een klant (naam, emailadres, adres) hotelkamers (kamernummer, soort bed (enkel/dubbel/stapel), tv aanwezig) kan reserveren. Deze reservatie is vanaf een bepaalde datum, tot een bepaalde datum, voor een bepaald bedrag.

Voor de klant laat je door Entity Framework zelf een eigenschap vinden die als primaire sleutel kan dienen. Een hotelkamer heeft diens nummer als primaire sleutel, en een reservatie heeft een primaire sleutel samengesteld uit de primaire sleutel van de klant, het kamernummer en de begindatum van de reservatie.

Als je de naam van een klant ingeeft, zie je zijn lopende reservaties (d.w.z. de reservaties die niet reeds zijn afgelopen).

Je haalt de data uit de databank d.m.v. eager-loading.

Je bouwt deze applicatie via *Three tier architecture*.

/ 30

/ 6

/ 8

/ 3

/ 3

/ 10



Vraag 5:

Maak een console applicatie die jouw diploma's kan bijhouden. Een diploma heeft een titel, onderwijsinstelling (string) en een jaartal waarin dit diploma werd behaald.

Bij het opstarten van de applicatie krijg je jouw diploma's te zien. Als je een 'X' intypt, kan je de titel van een diploma intypen. Dan wordt dit diploma verwijderd. Als je een 'V' intypt, kan je een extra diploma ingeven.

Deze data wordt opgeslagen in een JSON-bestand.

Je bouwt deze applicatie via *Three tier architecture*.

Data inlezen uit een bestand kan met:

*File.ReadAllText()*

Data schrijven naar een bestand kan met:

*File.WriteAllText()*

/ 25

/ 3

/ 6

/ 6

/ 10