

NAAM																			
VOORNAAM																			
KLAS																			
STUDENTNUMMER	0																		

Departement Wetenschap en Techniek

EXAMEN:

OPLEIDINGSONDERDEEL	Python development		
DEELOPLEIDINGSONDERDEEL OF FACET	Vaardigheidstoets		
LECTOR(EN)	Kristof Michiels		
OPLEIDING	EA-ICT / TI	OPLEIDINGSJAAR	2
ACADEMIEJAAR	2021 – 2022	EXAMENPERIODE	1
DATUM	22/6/2022	EXAMENDUUR	2u
WERKINSTRUMENTEN	Dit examen is open boek. Je mag online opzoeken. Je mag gebruik maken van papieren of digitale samenvattingen en eerder gemaakte oefeningen. Je mag tijdens deze toets op geen enkele manier communiceren met medestudenten of anderen. Je gebruikt Visual Studio Code of een editor naar keuze. Het gebruik van GitHub Copilot of een alternatief is evenwel niet toegestaan.		
RESULTAAT	. . / 20		

Inleiding

Je maakt voor deze test gebruik van Examon. Installeer deze tool vooraleer aan het examen te beginnen. Je vindt de instructies hier: <https://examonitor.ap.be>. Er mogen tijdens de toets geen vragen gesteld worden.

Je vertrekt van de bestanden die in de zip-file worden aangeboden en werkt in examen.py. Je volgt daarin de structuur die is voorzien. Jij behoudt de naamgeving en vult aan. Schrijf goed leesbare code en maak gebruik van correcte insprongen.

Na afloop van de toets stop je alle bestanden in een zipbestand met naam: **2122_PD_FamilienaamVoornaam_examen.zip** (JOUW familie- en voornaam!).

Belangrijk: je bent zelf verantwoordelijk voor:

- 1) indienen van de juiste bestanden
- 2) de goede werking van de ingediende bestanden
- 3) gebruik van de juiste naamgeving voor bestanden en zip-bestand
- 4) het op tijd indienen

Naam Student:

5) de goede werking van het zip-bestand waarin je de bestanden hebt verpakt (dus test nog even uit vooraleer in te dienen)

Je hebt 2u om aan de opdrachten te werken. Ik wens je veel succes!

Toepassing Drone Zilla	Score: /20
-------------------------------	-------------------

Algemene kadering:

Het bedrijf "Drone Zilla" is gespecialiseerd in het reinigen, controleren en onderhouden van zonnepanelen. Het bedrijf vraagt je een Python-toepassing te schrijven om de opdrachten voor hun drones te beheren. Je gebruikt hiervoor zoveel mogelijk de principes van object-georiënteerd design (klassen, methods, attributen, compositie en overerving). In bijlage vind je het bestand examen.py met al een aanwezige structuur. Je schrijft ALLE code in dit bestand. Bekijk ook onmiddellijk output.txt. Dit bestand toont de terminal-output waar je naartoe zal werken in je toepassing.

Drone Zilla beschikt over 20 drones. Er zijn 2 soorten: gewone drones (klasse Drone) en xl-drones (klasse DroneXL). De gewone drones kunnen één zonnepaneel servicen (capaciteit=1) en xl drones kunnen er twee servicen (capaciteit=2). Bij de start van je toepassing maak je 20 drone-objecten en je laat telkens het toeval beslissen hoeveel gewone en xl-drones er zijn.

Het bedrijf heeft recent 100 opdrachten binnengekregen. De opdrachten bestaan uit een familienaam en een aantal panelen dat moet worden onderhouden. Bij de start van je toepassing maak je 100 opdracht-objecten aan en je laat telkens het toeval beslissen hoeveel zonnepanelen elke opdracht bevat. Het aantal ligt daarbij telkens willekeurig tussen 1 en 5.

Je zorgt er voor dat de drones aan de opdrachten wordt toegekend. Je doorloopt alle opdrachten en bepaalt efficiënt welke drones de opdracht gaan uitvoeren. Met efficiënt wordt bedoeld: Een opdracht met 5 zonnepanelen bvb. kan worden uitgevoerd door drie gewone drones en 1 xl-drone, of door 1 gewone drone en 2 xl-drones, of door 5 gewone drones. Je controleert daarbij dat de drones allemaal zo gelijkwaardig mogelijk worden ingeschakeld.

Daarna rapporteer je naar de terminal een output die zo goed mogelijk lijkt op de tekst die je vindt in het bijgevoegde bestand output.txt. Tot slot schrijf je voor elke opdracht ook een regel naar een SQLite databank.

Deeltaken en puntenverdeling:

1. Je gebruikt de aanwezige methods in de main-functie om je deeltaken te vervullen. Je voorziet de aanwezige klassen van de nodige attributen en methods: Drone_zilla (een container-klasse voor de toepassing), Drone, DroneXL (erft zo efficiënt mogelijk van Drone) en Opdracht. Je spreekt instance-attributen niet rechtstreeks aan van buiten de klasse, steeds via methods (**4 punten** voor het algemeen goed invullen van de klassen).

2. Je zorgt voor het correct werken van de initialiseer_drones_en_opdrachten-method: je voorziet:

* 20 willekeurige drones (laat het toeval beslissen hoeveel gewone en hoeveel xl drones). Je zorgt voor een willekeurige naam en laat kiezen uit NAAMLIJST. Bij xl drones wordt aan de naam op het eind ook nog "XL" toegevoegd (werk hiervoor met een Class variabele).

* 100 opdrachten (willekeurig gekozen). De opdrachten bestaan uit een willekeurige familienaam en een aantal panelen dat moet worden onderhouden. Je laat telkens het toeval beslissen hoeveel zonnepanelen elke opdracht bevat. Het aantal ligt daarbij telkens willekeurig tussen 1 en 5.

(3 punten)

3. Je zorgt voor het correct werken van de opdrachten_en_drones_matchen-method. Je zorgt dat:

* elke opdracht een juist aantal drones krijgt toegewezen voor het aantal aanwezige zonnepanelen

* Alle combinaties mogelijk zijn: bvb een opdracht van 3 zonnepanelen kan door 3 gewone drones worden uitgevoerd, ofwel door 1 gewone en 1 xl-drone

* Je controleert dat de drones allemaal zo gelijkwaardig mogelijk worden ingeschakeld. Dus allemaal ongeveer een gelijk aantal keren

* Je houdt in attributen de nodige gegevens bij die nodig zijn voor de rapportering (zie volgende punt).

(6 punten).

- Je zorgt voor het correct werken van de rapporteur_naar_terminal-method. Je rapporteert naar de terminal en bootst daarbij zo goed mogelijk de tekst na die je vindt in het bestand output.txt. Dit bestand bevat een overzicht per opdracht en een overzicht voor de drones. Deze informatie wordt binnen de rapporteer-method verzameld uit object-data. Je mag de output-data dus niet aanmaken als string tijdens het afleveren van de pakketten. Je slaat de nodige data wel op in relevante object-attributen **(4 punten)**.

- Je zorgt voor het correct werken van de schrijf_weg_naar_db_method. Voor elke opdracht schrijf je een record met informatie weg naar 1 tabel (opdrachten) in examen.db (SQLite) die bestaat uit een id (int), naam (text), aantal_panelen (int) en drone_lijs (tekst) **(3 punten)**