

# Project tweede examenkans: BroodCo

Het bedrijf BroodCo, dat een aantal broodautomaten in de provincie Antwerpse uitbaat, benadert je om hun toestellen naar een volgend - slim - niveau te brengen. Ze hebben hun automaten verbonden met het internet en voorzien van de nodige sensoren. De toestellen kunnen hierdoor hun status meegeven. Dit stelt het bedrijf in staat om korter op de bal te spelen: gericht te produceren, klanten te volgen, en de toestellen “slimmer” te laten bijvullen door koeriers.

Je maakt een toepassing in Python en gebruikt zoveel mogelijk de principes van object-georiënteerd design (klassen, methods, attributen, compositie en overerving). Je werkt volgens eigen keuzes en voorkeuren, maar houdt wel rekening met de hieronder opgesomde vereisten:

- Startbestand heet **main.py** en je splitst de toepassing zo helder mogelijk op in verschillende bestanden. Je zorgt hierbij voor een passende maar logische naamgeving.
- Bij het uitvoeren van de opdracht voorzie je de klassen op een zo logisch mogelijke manier van properties en methods.
- Je gaat voor het bijhouden van geïnstantieerde objecten zo veel mogelijk gebruik maken van object compositie. Je houdt de objecten telkens bij in een dictionary property: bvb. BroodCo heeft automaten, automaten hebben broden, klanten hebben broden, koeriers hebben broden, ...
- Je creëert klassen voor de volgende entiteiten: BroodCo, BroodAutomaat, GroteBroodAutomaat (erft van BroodAutomaat), Brood (een Abstract Base klasse), KleinBrood (een Abstract Base klasse, erft van Brood), GrootBrood (een Abstract Base klasse, erft van Brood), VolkorenBrood (erft van GrootBrood, WitBrood (erft van GrootBrood), MeergranenBrood (erft van GrootBrood), ZuurdesemBrood (erft van KleinBrood), RozijnenBrood (erft van KleinBrood), Klant, Koerier.
- Bij de start van de toepassing instantieer je een BroodCo object, waarbij het bedrijf automatisch van 25 automaat-objecten wordt voorzien: 20 gewone en 5 grote. Elk automaat heeft een capaciteit. De capaciteit van een gewone is 75

grote broden. De capaciteit van een groot automaat is 105 grote broden. Kleine broden nemen de helft van de plaats in van grote broden. Je voorziet voor de automaten methods om een brood te kopen (door klanten), toe te voegen (door koeriers), te verwijderen (door koeriers). Je zorgt ervoor dat bij het toevoegen de capaciteit niet kan worden overschreden.

- Je voorziet ook een initieel bakmoment waarbij 300 broden van elk van de vijf types worden gebakken. Dus 1500 in totaal. Je voorziet hier een specifieke method, die zo herbruikbaar mogelijk is voor bakmomenten met andere aantallen en types brood. Je zorgt dat het tijdstip van bakken wordt toegevoegd aan elk brood.
- Na het bakproces worden de broden verdeeld over de automaten door koeriers. Je maakt 5 koeriers aan die elk 300 willekeurige broden over de automaten distribueren. Je kiest hier zelf hoe je dit proces aanpakt, maar de capaciteit van een automaat mag niet worden overschreden.
- Je voorziet en gebruikt methods om na het initialisatieproces op heldere wijze de status van elk van de automaten naar de console af te drukken. Ook de gebruikte en nog beschikbare capaciteit dienen te worden weergegeven.
- Je creëert in je code een 5-tal klanten en gebruikt de in de Automaat-klasse gecreëerde method om voor elke klant een brood te kopen aan een bepaald automaat.
- Je zorgt voor een manier waarbij bij een klant wordt bijgehouden welk brood, wanneer en aan welk automaat werd aangekocht. Hier zou in een latere fase een klantenkaart kunnen worden gekoppeld. Je print voor één klant deze informatie naar de console. Maak hiervoor gebruik van een specifieke geschikte method.