

Oppgave 4 Take-away tjeneste (45 poeng)

Du skal skrive (deler av) et program for en take-away tjeneste som tar imot bestillinger fra kunder på internett, gjennom et enkelt, terminalbasert grensesnitt. Programmet skal holde rede på en rekke faste kunder med telefonnummer og hvilke typer mat/ ingredienser (*innholdsstoffer*) denne kunden *ikke* ønsker (for eksempel gluten, meieriprodukter eller svinekjøtt).

Videre skal programmet kjenne til hvilke matretter som kan leveres innenfor en rekke *kategorier* – som for eksempel forretter, hovedretter, eller desserter. For hver rett skal det lagres hvilke innholdsstoffer retten har som kan være problematiske for kunder. I denne oppgaven kan du anta at ordene som blir brukt om innholdsstoffer kunden vil unngå, og ordene som blir brukt om innholdstoffene i en rett, hentes fra samme vokabular slik at det er enkelt å sammenligne.

Programmet bruker disse dataene til å presentere en tilpasset meny til hver kunde, der alle matrettene i menyen tilfredsstiller kundens krav til innhold.

Den delen av systemet du trenger å kjenne til består foruten **TakeAway** av klassene **Rett**, **Kategori**, **Meny** og **Kunde**. Du skal skrive alle metoder som er beskrevet for hver klasse om det ikke eksplisitt er oppgitt at de ikke skal skrives. Det anbefales å lage en tegning av strukturen for eget bruk.

a) 10 poeng. Skriv klassen **Rett**

Klassen har et navn (streng), en pris (flyttall), og en liste (kan være tom) av innholdsstoffer (strenger). Alle instansvariable får startverdi fra parametere til konstruktøren `__init__`. Klassens grensesnitt har følgende metoder i tillegg til konstruktøren:

- **sjekkInnholdOK** har en parameter med en liste av innholdsstoffer. Metoden sjekker om noen av innholdstoffene i parameteren fins i rettens liste over innhold. Hvis metoden finner et treff, returnerer den False. Hvis ingen av innholdstoffene i parameteren finnes i retten, skal metoden returnere True
- `__str__` returnerer en streng med rettens navn, pris og alle innholdsstoffer på en lesbar form.

b) 5 poeng. Skriv klassen **Kategori**

Klassen har to instansvariable; kategorinavn og en liste med referanser til **Rett**-objekter. Begge får verdi fra parametere til konstruktøren. Grensesnittet til klassen Kategori inneholder dessuten følgende metode:

- **hentOkRetter** med én parameter: En liste med innholdsstoffer som en kunde ønsker å unngå. Metoden går gjennom kategoriens liste over retter, og lager en ny liste med referanser til retter som ikke inneholder noen av innholdstoffene som skal unngås. Denne nye listen returneres. (Du trenger ikke lage kopier av Rett-objektene som skal være med i den nye listen, kun referere til de samme objektene som instansvariabelen).

c) 10 poeng. Skriv klassen **Meny**

Klassen har én instansvariabel. Denne representerer hele menyen i en ordbok med alle kategorinavnene som nøkler og referanser til Kategori-objekter som verdier. Klassens konstruktør har én parameter; en liste med alle kategorinavn i menyen. Alle data om en kategori leses fra en fil der filnavnet består av kategorinavnet etterfulgt av ".txt". Konstruktøren skal bygge opp menyen ved å kalle på den private (non-public) metoden `_lesKategoriFil` som har et filnavn som parameter og returnerer et ferdig kategoriobjekt. Du skal *ikke* skrive metoden `_lesKategoriFil`.

Foruten konstruktøren har klassen én metode:

- **hentRedusertMeny** tar én parameter; en liste over innholdsstoffer som skal unngås. Metoden går gjennom hele menyen kategori for kategori, og bygger opp en annen, redusert meny der ingen av rettene inneholder uønskede innholdsstoffer. Kategorier som ikke har noen retter igjen skal ikke være med i den reduserte menyen som returneres av metoden.

d) 8 poeng. Skriv klassen **Kunde**

Klassen har to instansvariable: Telefonnummer til kunden (en streng), og en liste med innholdsstoffer (strenger) som kunden ønsker å unngå (for eksempel på grunn av allergi). Begge instansvariable får verdier fra parametere til konstruktøren. Videre har klassen en metode

- **velgRetter**, som tar et Meny-objekt som parameter og kaller på `hentRedusertMeny` på dette objektet for å få en skreddersydd meny for kunden. Deretter presenteres den reduserte menyen på terminalen for kunden, en kategori av gangen. Kunden velger en rett fra kategorien ved å taste inn navnet på retten – eller en tom linje for å hoppe videre til neste kategori. Hver ikke-tomme linje tastet inn av kunden lagres som en streng, og metoden returnerer disse strengene i en liste. Kundens input skal altså ikke sjekkes mot retter i systemet – dette for at kunden kan legge inn egne beskjeder til kjøkkenet om mengde eller tilberedning.

e) 7 poeng. Skriv klassen **TakeAway**

Klassen **TakeAway** har to instansvariable; en referanse til et objekt av klassen Meny, og en ordbok med kunder der telefonnummer er nøkkel og verdiene er referanser til Kundeobjekter.

Konstruktøren har to parametere; en liste av kategorinavn og navn på en kundefil, og skal bygge opp en ferdig meny en og kundekatalog. Metoden `_lesKundefil` returnerer en kundekatalog med alle kunder og skal kalles i konstruktøren for `TakeAway`, men *ikke* skrives av deg. Foruten konstruktøren er grensesnittet til klassen som følger:

- **betjenKunde** med parameter telefonnummer for en kunde som har tatt kontakt. Metoden kaller på metoden `velgRetter` for riktig kunde (du kan anta at alle kunder er registrert på forhånd). Deretter kalles den private (non-public) metoden `_lagOgLeverMat` med bestillingen fra `velgRetter`. `_lagOgLeverMat` har en parameter bestilling og skal i denne versjonen kun skrive ut kundens bestilling (navn på alle rettene) på terminalen. Metoden `_lagOgLeverMat` skal skrives av deg.

f) 5 poeng. Skriv et hovedprogram

Hovedprogrammet skal gjøre følgende:

- Starte en take-away tjeneste med kategoriene "Forretter", "Hovedretter" og "Desserter" på menyen, og en kundekatalog på filen "Kunder.txt". Du kan anta at alle nødvendige datafiler finnes.
- Be om telefonnummer (på terminalen) for, og betjene, en og en kunde inntil bruker gir inn en tom streng som telefonnummer - dette avslutter programmet.