

Inlämningsuppgift Python programmering

Januari 2022

Generell information

Inlämning sker i form utav Python-filer (.py). Zip ihop alla filer i en mapp och lämna in på PingPong. Deadline 2022-02-04 kl 23:55. Den zippade mappen namnges med namn, Python och betygsönske, tex *Eva_Hegnar_Python_VG.zip*.

Kom ihåg att kopiera kod rakt av räknas som fusk, men man får hämta inspiration. Inlämningen är **individuell**.

Betygskriterier

G

- På ett grundläggande sätt kunna redogöra för programmeringsspråket Python och dess syntax
- På ett grundläggande sätt kunna förklara klasser och objekt i Python
- På ett grundläggande sätt kunna förklara databashantering i Python
- På ett grundläggande sätt kunna förklara Python script
- På ett grundläggande sätt kunna organisera Python projekt med paket och klasser
- På ett grundläggande sätt kunna hantera filer med Python
- På ett grundläggande sätt kunna skapa och analysera Python projekt
- På ett grundläggande sätt kunna skapa kod i Python till en relationsdatabas
- På ett grundläggande sätt kunna tillämpa uttryck funktioner, variabler och datatyper i Python

VG

- Uppnått kraven för betyget Godkänd
- På ett självständigt sätt kunna organisera Python projekt med paket och klasser
- På ett självständigt sätt kunna hantera filer med Python
- På ett självständigt sätt kunna skapa och analysera Python projekt
- På ett utförligt sätt kunna skapa kod i Python till en relationsdatabas
- På ett utförligt sätt kunna tillämpa uttryck funktioner, variabler och datatyper i Python

Introduktion

I inlämningsuppgiften ska du skapa ett program som läser in ett dataset till en relationell databas. Du ska förändra rådata till användbar data och sedan analysera och visualisera datan. Datan du ska använda är vaccinationstakten mot Covid-19 för olika länder i världen. Din kod ska skrivas på ett objektorienterad sätt och ska innehålla båda privata och publika metoder.

G

För godkänt ska du skapa en databas och ladda in en csv-fil i databasen. Du ska även städa datan. Du ska programmera objektorienterad med en main-metod och en klass.

- Det ska finnas två pythonfiler, en som innehåller main-metoden (ingången till programmet) samt en som innehåller datahanterings-klassen.
- Main-metoden ska läsa in csv-filen och skapa och använda datahanterings-klassen för att utföra sitt arbete.
- Datahanterings-klassen har metoder för att skapa en databas och fylla på med csv-data. Programmet ska kunna läsa in datafilen *vaccin_covid.csv*. Datan ska sparas i databasen med hjälp av SQLite. Klassen behöver alltså de två publika metoderna, till exempel *create_database()* och *seed_database(path_to_file)*.
- Datahanterings-klassen ska också ha en normaliserings-metod. Du behöver städa datan. Kom ihåg normalisering av databaser från föreläsning 8 i SQL-kursen. Här pratade vi om "The three normal forms" som en relationell databas måste uppfylla. Du ska uppfylla First normal form "The data are stored in a two-dimensional table" och att "There are no repeating groups". Det vill säga att varje cell ska bara ha ett värde. Detta är inte fallet i kolumn *vaccines*. Här ska du använda SQL-kommandon genom Python för att ändra i tabellen.
- Det finnes rätt mycket NaN värden som måste hanteras. Är det ointressanta kolumner som kan exkluderas? Kan man sätta ett annat värde i stället för NaN?
- Sammanfattningsvis kommer det vara en main-fil och en klass-fil med alla metoder. Tips på handlingsförlöpp i main-filen kan vara att först skapa ett objekt av klassen, sedan läsa in csv-filen (metod), skapa en ny databas (om den inte finns) (metod), ladda in data i databasen (metod) och normalisera datan i SQL (metod).
- Kom ihåg programmerings "conventions" från föreläsning 6. Ge meningsfulla variabelnamn och använd docstrings för att förklara vad alla metoder gör.

VG

För VG ska du uppfylla kraven för godkänt. Du ska också dela databasen i kopplade tabeller och extrahera information från databasen.

- Du ska manipulera datan ytterligare så att den blir en relationell databas. Databasen ska uppfylla Second och Third Normal form: "All non-key attributes are functionally dependent on the entire primary key" och "There are no transitive dependencies". Du ska dela datan upp i kopplade tabeller och använda primary och foreign keys. Detta ska göras genom metoder för SQL i Python.
- Skapa en ny klass med privata och publika metoder för att analysera och visualisera datan. Ett förslag på privat metod kan vara *extract_country_data(country)* som returnerar ett subset av all data med hjälp av SQL-kommandon. Ett förslag på en publik metod kan vara *plot_daily_vaccinations()*.