Kommentarer til klassenavn:

Jeg foretrekker skrivemåten ‘Todo’ over ‘ToDo’, fordi det er den vanligere måten å skrive det på. Dette betyr at jeg ville ha endret klassenavnene ‘ToDoListApp’ og ‘ToDoList’ til ‘TodoListApp’ og ‘TodoList’ henholdsvis.

Kommentarer til klassevariabler:

Klassevariabler burde for det meste være private. ‘root’ og ‘toDoList’ variablene i ‘ToDoListApp’ burde være ‘\_\_root’ og ‘\_\_toDoList’.

Kommentarer til metoder:

Metoder som ikke brukes eksternt burde være protected. Det virker som om ‘create\_gui’ er en intern metode i ‘ToDoListApp’ klassen siden den blir kalt i ‘\_\_init\_\_, så denne metoden burde være protected og dermed hete ‘\_create\_gui’. Metoden burde egentlig ikke kalles i ‘\_\_init\_\_’, fordi constructors ikke skal ha side effects som for eksempel å starte en uendelig løkke slik som skjer når man bruker tkinter. Metoden burde kalles gjennom en annen public metode som kan hete ‘run\_gui’ eller noe lignende.

Metoden i ‘ToDoList’ som returner oppgavene (‘Tasks’) følger ikke «snake\_case naming convention» som funksjoner og metoder skal følge i Python. Den burde også ha et navn som formidler funksjonaliteten bedre, altså ville et bedre navn ha vært ‘get\_tasks’ eller ‘get\_todos’ for å være konsekvent med hvilket substantiv som blir brukt om oppgavene.

Kommentarer til videreutvikling:

For å støtte filtrering kan man oppdatere metoden ‘get\_todos’ (tidligere ‘Tasks’ ☹) til å akseptere to ekstra parametere ‘user\_id’ og ‘only\_show\_completed’. Med disse parameterne kan metoden filtrere ut oppgavene som ikke fyller kravene og returnere resten. Koden til ‘\_create\_gui’ trenger ikke å endres stort, vi må bare passe på å gi de riktige parameterne til metoden. Verdiene ‘user\_id’ og ‘only\_show\_completed’ burde være tilstandsvariabler som blir oppdatert gjennom de interaktive gui-elementene. Resten av koden kan forbli slik som den er.