

Peer review

Joel Henriksson, peer review av Anton Gong

December 8, 2024

1 Projektinformation

Den här texten är en granskning av **Parkeringshuset** uppgiften, skapat av **Anton Gong**. Projektet kommer att granskas utifrån kriterierna för god programmeringspraxis från uppgiftsinstruktionerna.

2 Granskning enligt programmeringspraxis

2.1 Struktur

Koden är väl strukturerad och uppdelad i klasser, metoder och funktioner. Det finns tydliga avgränsningar mellan logiken för hanteringen av data och tkinter grafiken.

2.2 Lagom stora funktioner, klasser och metoder

Funktionerna och metoderna är av lagom storlek, är enkla att förstå och har en specifik uppgift.

2.3 Dokumentation

Koden innehåller många kommentarer som är lätta att förstå och kortfattat beskriver olika delar av koden. Det finns inga övergripande kommentarer för varje funktion och klass som sammanfattar in- och returvärden, men jag tycker inte att detta är något som nödvändigtvis behövs eftersom det istället finns flera mindre kommentarer inom varje funktion och klass som beskriver hur de fungerar. Om du ändå vill lägga till fler kommentarer kan du använda den här guiden för docstring.

2.4 Lokala variabler

Lokala variabler används konsekvent inom funktioner och klasser. Istället för att använda globala variabler så är vissa värden som används upprepade gånger inom olika funktioner av samma klass inpackade i klassen som fält.

2.5 Kodupprepning

Det förekommer ytterst lite kodupprepning och alla av programmets huvudsakliga operationer är inskrivna i egna funktioner som används många gånger genom hela programmet, vilket undviker upprepning och gör koden mer kompakt och läsbar.

2.6 Återanvändbar och utvidgningsbar kod

Programmet är välstrukturerat och utformat för att vara utvidgningsbart. Ett särskilt starkt exempel är hanteringen av biltyper och deras tillhörande attribut, såsom timkostnaden. Denna attribut implementeras på ett sätt som gör det enkelt att lägga till fler bilkategorier med olika egenskaper i framtiden.

2.7 Användargränssnitt och felhantering

Användargränssnittet är enkelt och lätt att förstå. Om användarens input av data är inkorrekt på något sätt, så ger programmet ut ett pedagogiskt och lättförståeligt grafiskt meddelande som beskriver hur användaren bör åtgärda problemet. Till exempel, om användaren försöker ge ett registreringsnummer som inte är 6 siffror långt, så ges detta felmeddelande (i grafiska användargränssnittet): "Registreringsnummret måste vara 6 tecken långt".

Jag har dock hittat en bug i användargränssnittet som gör att man inte kan registrera mer än en bil vid varje körning av programmet. För att återskapa buggen gör så här:

1. Registrera en ny bil med registreringsnummer och storlek.
2. Tryck på "gå tillbaks till huvudmenyn."
3. Om man efter detta klickar på "registrera ny bil" tas man direkt till skärmen för storlek och kan inte ange ett nytt registreringsnummer.

2.8 Variabelnamn och läsbarhet

Variabel- och funktionsnamnen är i överlag beskrivande och lätta att förstå. Det finns dock vissa undantag, såsom `True_tal` i `Parkering_tkinter`, som till exempel skulle kunna ändras till `bil_hittad` för ökad läsbarhet.

3 Sammanfattning och rekommendationer

Det här programmet och koden som utgör det är ytterst välskrivet och uppfyller tydligt de åtta punkterna för god programmeringspraxis, bortsett från två små anmärkningar:

1. Fixa buggen som gör att man inte kan lägga till mer än en bil inom samma körning av programmet.
2. Ändra på några variabelnamn, som till exempel `True_tal` i `Parkering_tkinter`.