

Kassa

en enkel kassa mjuvvara

Sam Sadeghi SY17

2019/2020

Abstract

The software for the cash register at my uncle's grocery store is slow and I believe I can make a program that is faster and runs smoother than the software found on the cash register at my uncle's grocery store. In creating this program i will learn to write in the programming language python and how to make graphical apps using the graphics library TKinter for python. I create a nice looking GUI (graphical user interface) made of buttons and labels and a lot of functions to make everything work. Due to poor time management and lots procrastinating the program ends up not having all the features i had wished for but all in all it is alright because I am smarter now than I was when I began this project

Innehåll

1	Inledning	3
1.1	Syfte	3
1.2	Mål	3
1.3	Metod	3
2	Utförande	5
3	Resultat	8
4	Diskussion	9
5	Slutsats	10

1 Inledning

Ibland hjälper jag min farbror i hans livsmedelsbutik. Där får jag använda mig av kassaapparaten. Programmet som kassaapparaten på farbrors butik kör är långsamt och jag tänkte att jag skulle kunna göra ett liknande program som kör snabbare genom att endast ha med de nödvändigaste funktionerna.

1.1 Syfte

I skapandet av det här programmet ska jag lära mig hur man gör ett grafiskt program med hjälp av programmeringsspråket Python och gränssnittsverktyget Tkinter.

1.2 Mål

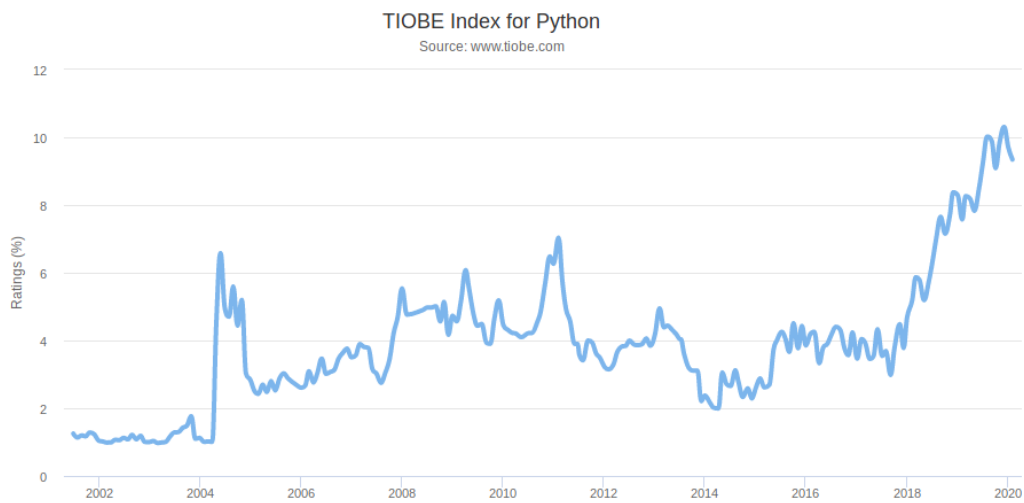
Vid slutet av arbetet ska jag ha ett enkelt kassa program som kan räkna ihop summan av flera varor samt hur mycket moms som skall betala beroende på varans vad för slags vara det är. Programmet skall även kunna skriva ut ett kvitto med tid datum vad man har köpt samt netto och brutto pris. Man ska kunna scanna varor som tidigare har repriserats. Programmet ska kunna anpassas så den kan användas av olika sorters affärer eller restauranger.

1.3 Metod

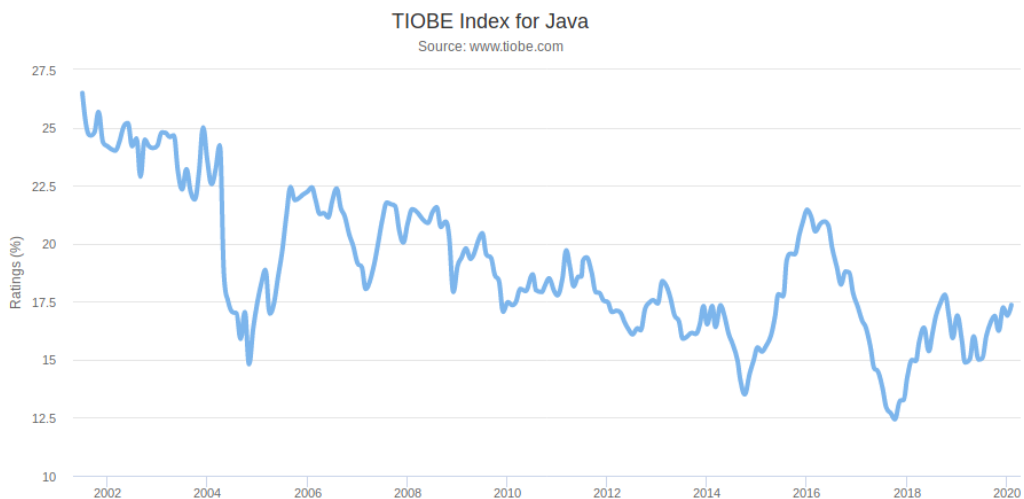
Redskap jag kommer använda är

Vim:

Vim är ett textredigeringsprogram som körs i konsolen. Det finns massor med fördelar med Vim. En av dessa fördelar är att den inte tar lika mycket resurser att köra Vim som till exempel Visual Studio men det i sig är inte så viktigt utan den största anledningen till varför jag jag använde Vim för att skriva det här programmet är att jag har vant mig vid Vims massor med tangentbordsgenvägar och det hjälper mig skriva snabbare inte bara det utan jag är också väldigt bekväm med att konfigurera Vim så att lättare kan skriva. Ett av dessa konfigurationer jag har gjort för att underlätta mitt skapande av programmet är att jag har gjort så att när jag trycker ner på **Ctrl** och **W** samtidigt så startar den programmet. Andra program kan säkert göra någonting liknande eller kanske till och med kommer inbyggt med en sådan funktion men i Vim får jag göra det själv som får mig att bättre förstå vad som händer i bakgrunden och som sagt tidigare är jag mycket bekvämare med Vim än vad jag är med andra textredigeringsprogram



Figur 1: TIOBE index för python



Figur 2: TIOBE index för java

Python:

Python är ett programmeringsspråk. Jag valde just Python för att skriva det här programmet för att jag ser det väldigt ofta i min omgivning. Enligt TIOBE indexet [2] är Python det tredje populäraste språket och dess popularitet verkar öka (se figur 1) till skillnad från språket som ligger på första plats det vill säga Java (se figur 2) och ju populärare ett programmeringsspråk är så finns det fler projekt som är skrivna i de programmeringsspråket därför tycker jag att det är värt att lära sig /skriva i Python.

Tkinter/TK:

Tkinter är en API för att använda TK i Python. TK är grafiskt gränssnittsverktyg som fungerar på Windows, MacOS och Linux. Jag valde Tkinter på grund av att den fungerar på dom tre vanligaste operativsystemen. Tkinter kommer

FPDF:

FPDF är ett programbibliotek för att generera PDF filer. PDF filer är dokument filer som är optimerade för att skrivas ut.

Github och Git:

Github är en webbserver med fokus på programutveckling. Github använder sig av version kontroll programmet Git. Git gör det enkelt för mig att se vad jag har gjort och när jag har gjort det. Git tillsammans med Github låter mig jobba på programmet var jag än är utan att vara uppkopplad till internet

Tutorialpoints sida på Python GUI Programming:

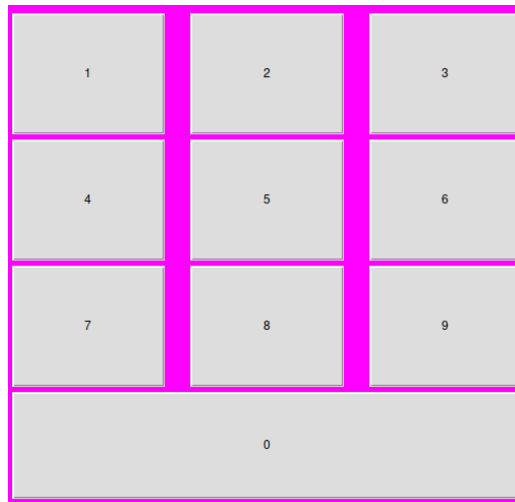
Tutorialpoints sida på Python GUI Programming är en webbsida som har i syfte att lära hur man använder TKinter. Jag valde att använda Tutorialpoints sida istället för den officiella TKinter manualen eftersom Tutorialpoints sida är mycket enklare att hitta information på. Det finns en lista på alla Widgets och en kort beskrivning av vad dem används för; du kan klicka på namnet av en widget du vill veta mer om och du tas till en sida där du får veta syntaxen och alla tillval för widgeten du klickat på.

2 Utförande

När jag börja projektet hade jag aldrig tidigare skrivit program i Python och det enda grafiska gränssnitt jag tidigare använt var HTML på Webben så det första jag behövde göra var att lära mig. Youtube videon "How to Program a GUI Application (with Python Tkinter)!" av Keith Galli[1] användes för att få en förståelse av hur man gör grafiska program med Tkinter och python.

Först behövdes ett sätt att skriva in siffror i programmet. Man skulle kunna använda nummer knapparna på tangentbordet men eftersom programmet ska kunna användas med endast en pekskärm så behövdes nummer knappar på skärmen, Detta gjorde jag genom att skapa tio knappar (från 0 till 9) och placera dem i ett 3 gånger 3 rutnät med en stor knapp för noll under det (se figur 3).

Därefter behövdes det ett sätt att se vad för nummer man skrivit; för det användes det en label som döptes till text tillsammans med en global variabel som deklarerades med namnet num. Num är noll när programmet startar men när en nummer knapp trycks så körs en funktion vid namnet knapp_tryck med värdet på knappen som ändrar num. Funktionen knapp_tryck fungerar genom att multiplicera num med tio, addera värdet den fick in till num och



Figur 3: hur nummer knapparna ser ut

tillslut visa det nya värdet för num på text. Nu kan användaren skriva in siffror i programmet men det saknas ett sätt att ta bort eller ångra siffror man skrivit därför implementerades en knapp som raderar den sista siffran på talet och en knapp som tar bort hela talet. Knappen som tar bort sista siffran är märkt med symbolen "<" och fungerar genom att kalla på funktionen knapp_del. Funktionen knapp_del fungerar på så sätt att den konverterar num till en sträng (detta görs för att det är lättare att kolla storleken och ta bort sista karaktären av en sträng). När num har konverterats till en sträng så kontrolleras det om num har fler än en karaktär i så fall tas den sista karaktären bort. Om num bara har en karaktär när knapp_clear kör kommer num sättas till noll. Knappen som tar bort hela talet har bokstaven "C" på sig och kallar på funktionen knapp_clear när den är tryckt. Knapp_clear är en funktion som byter värdet av num till noll.

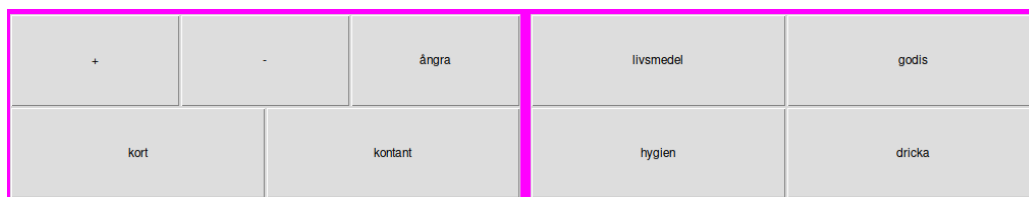
Nu när det finns nummer knappar och ett sätt att se vilket nummer man har skrivit då behövs det ett sätt att säga till programmet vad numret betyder. Numret som har skrivits in kan vara en av tre saker. Det kan vara en produkt, en betalning eller ett antal.

En produkt är något som kunden ska köpa. Produkter representeras i koden med klassen prod. Prod har attributen namn, antal, pris och moms och en funktion som heter totpris som ger priset multiplicerat med antalet. Man väljer att numret är en produkt genom att trycka på en av produkt knapparna (se figur 4) efter att skrivit in ett nummer. Numret blir på priset av produkten.

En betalning är hur kunden betalar för produkten. Betalningar behandlas

av koden likt hur produkter behandlas det vill säga betalningar representeras av klassen betal som är precis som klassen prod med skillnaden att betal inte har moms attributen. Man säger till programmet att numret är en betalning genom att trycka på en av betalnings knapparna (se figur 4) efter man har valt ett nummer. Numret blir hur mycket som ska betalas med valt betalningsalternativ. Om numret är noll kommer knappen att bli grön när och när man trycker på en grön knapp så lägger programmet en betalning som är lika stort som total priset för alla produkter minus alla tidigare betalningar.

Antal är som sagt tidigare en attribut hos produkter och betalningar. Antal för den senaste produkten eller betalningen kan ändras men plus och minus knapparna (se figur 4). När man trycker på en av antals knapparna så kollar programmet först om num är noll. Ifall num är noll så adderas eller subtraheras (beroende på om plus eller minus knappen trycktes) en från senaste produkten eller betalningens antal, men om num är över noll så kommer num adderas eller subtraheras från senaste produkt eller betalning.



Figur 4: Till höger är dom fyra produkt knapparna markerade som livsmedel, godis, hygien och dricka. Till vänster är dom två betalnings knapparna markerade med orden kontant och kort. Ovanför betalnings knapparna är antals knapparna märkta med "+" och "-". Bredvid antals knapparna finns ånga knappen.

Om man vill ta bort den senaste produkten eller betalningen man har skapat finns det en ånga knapp. Ånga knappen fungerar genom att köra funktionen knapp_remove. Alla betalningar och produkter sparas i en global array (en array är en lista i programmet) som heter kvitto. Vad funktionen knapp_remove gör är att den tar bort det sista objektet från arrayen kvitto.

För att kunna se alla betalningar och produkter skapas en tabell med hjälp av labels. Tabellen har fyra kolumner som har rubrikerna antal, namn, pris och pris f.a. (pris för alla). Tabellen fungerar genom att funktionen update_display fyller tabellen med relevant information från objekten i arrayen kvitto. Funktionen körs varje gång ett objekt läggs till i arrayen kvitto eller tas bort från arrayen kvitto.

Under tabellen skapades en lable där det står hur mycket allt kommer kosta totalt. Om priset för alla produkter sammanlagt är mer än "priset"

antal	namn	pris	pris f.a.
2	livsmedel	19	38
1	dricka	12	12
2	godis	5	10
totalt: 60 kvar att betala: 60			

Figur 5: hur tabellen ser ut när den används

för alla betalningar så ska det stå hur mycket det är kvar att betala bredvid totala kostnaden men om motsatsen är sann kommer köpet vara klart och kvar att betala ändras till växel.

När köpet är klart så skapas en PDF fil med kvittot på det du köpt i en mapp med dagens datum och sedan töms arrayen kvitto och programmet är redo för ett nytt köp.

3 Resultat

Resultatet av arbetet är ett enkelt program för som kan lätt användas för att skapa kvitton. Programmet är gjort för att användas i affärer av en kassör eller vart som helst där köper saker i person och kan behöva ett kvitto.

Att använda programmet är inte helt självklart när man först kollar på det men det är inte svårt att lära sig. Om en kund ska köpa något så skriver man först in hur mycket det kostar med hjälp nummerknapparna på skärmen och därefter trycker man på knappen som bäst beskriver vad kunden har köpt. Om kunden har köpt fler än en av samma sak kan plus och minus knapparna användas för att öka eller minska antalet om antalet är mycket kan man skriva hur många gånger man ska trycka på antingen plus eller minus knappen med nummerknapparna sedan trycka på önskad knapp. Om kunden ångrar sig eller om kassören har gjort ett misstag kan man ångra genom att trycka på

antal	namn	pris	pris f.a.				
				0		<	C
				1	2	3	
				4	5	6	
				7	8	9	
				0			
totalt: 0							
+		-		ångra			
kort		kontant		livsmedel		godis	
				hygien		dricka	

Figur 6: hur det färdiga programmet ser ut

ångra knappen. Slutligen trycker man på antingen kort eller kontant knappen beroende på hur kunden vill betala. Om kunden vill betala delvis med kort och delvis med kontant kan man skriva hur mycket kunden vill betala med kontant innan man trycker på kontant knappen, detsamma gäller för kort knappen

4 Diskussion

På grund av dålig tidsplanering och stora mängder prokastinering blev slut produkten inte precis så jag ville ha det.

Programmet kan inte skanna streckkoder eftersom jag kunde inte komma på ett smidigt sätt skanna streckkoder. Jag hade inte heller tillgång till en streckkod-scanner vilket gjorde mig osäker på hur streckkod-scanner fungerade och om idéerna jag hade på hur jag skulle implementera scannandet av streckkoder skulle fungera. Allt det plus att jag började få ont om tid fick mig att ge upp på att programmet ska kunna skanna streckkoder.

Jag hade inte en kvitto-skrivare och det är därför programmet inte skriver ut kvittona utan bara skapar en pdf fil för att skrivas ut; jag vet faktiskt inte om kvittoskrivare kan skriva ut pdf filer jag antog bara att efter som vanliga skrivare kan det kan nog kvittoskrivare det med.

I början av arbetet hade jag en ide om att man skulle kunna ställa in saker som namnet på toppen av kvittot, bredden på kvitto pappret, färgen

i bakgrunden, vad man kan köpa och hur man kan betala i en konfiguration fil vilket skulle göra programmet mycket enklare att anpassa. I nuläget kan man endast ändra dessa saker genom att ändra på koden men för att göra det krävs det grundläggande kunskaper om programmering och programmeringsspråket python vilket inte alla har och jag vill att programmet ska användas av vem som helst.

Syftet med arbetet var att jag skulle lära mig programmera i python och göra grafiska appar, saker som jag aldrig gjort förut, och nu när arbetet är klart kan jag med säkerhet säga att jag kan det. Jag har lärt mig mycket mer än vad jag trodde jag skulle lära mig. Jag visste om klasser sen tidigare men jag har aldrig riktigt förstått vad dom skulle användas för eller hur man använder dom men i skapandet av detta programmet förstår jag klasser väldigt mycket bättre än vad jag gjorde tidigare.

5 Slutsats

Jag lyckades kanske inte helt med mitt mål men jag kom rätt så långt och nu är jag mer kunnig än jag var när jag påbörjade det här arbetet. I framtiden kommer jag nog planera min tid bättre genom att till exempel sätta upp mål och deadlines för mig själv.

Vissa saker jag inte hann göra som exempelvis konfigurationsfilen är inte så svårt att fixa och kan lätt läggas till i nästa uppdatering av programmet men för just nu så fungerar programmet och den gör vad den är gjord för att göra.

Referenser

- [1] Keith Galli. *How to Program a GUI Application (with Python Tkinter)!* Youtube. 2019. URL: <https://www.youtube.com/watch?v=D8-snVfekto>.
- [2] TIOBE. *TIOBE Index for February 2020*. 2020. URL: <https://www.tiobe.com/tiobe-index/> (hämtad 2020-03-01).