BUDGET TRACKING APPLICATIE

OPGAVE

Maak een applicatie die de gebruiker helpt zijn/haar uitgaven en inkomsten bij te houden.

REQUIREMENTS

De gebruiker:

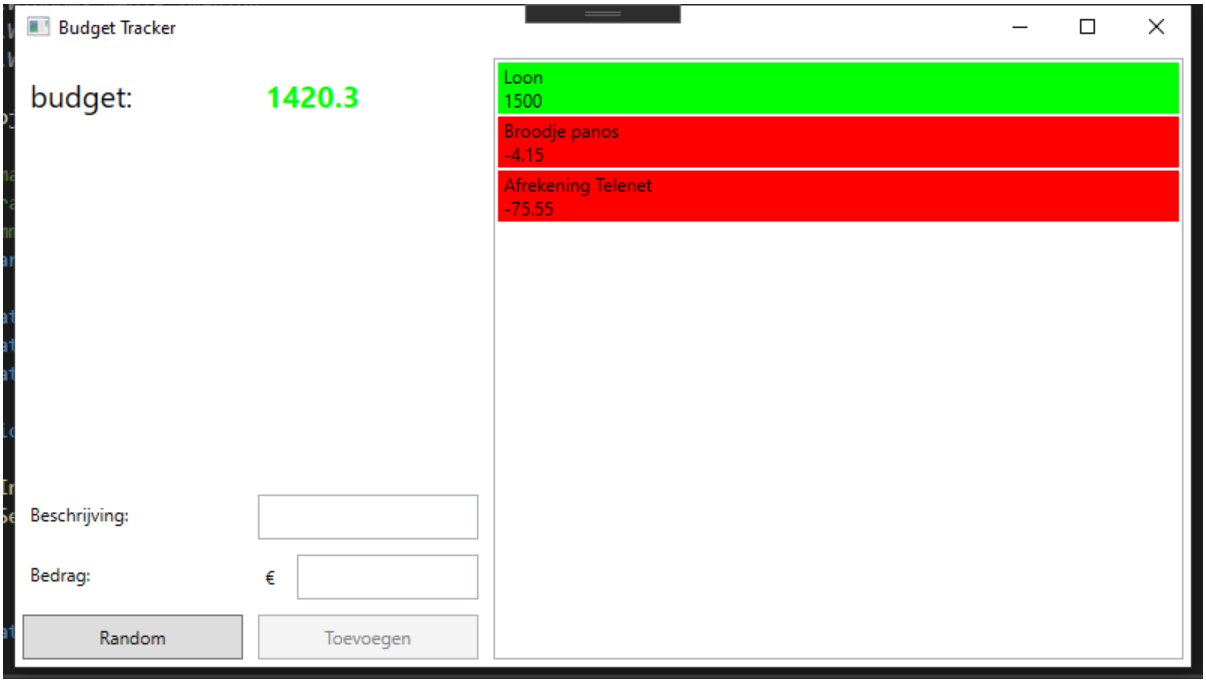
- Kan zijn huidige budget zien
    - Het budget is rood gekleurd wanneer dit negatief is
    - Het budget is groen gekleurd wanneer het positief is
- Kan sporadische inkomsten in geven
- Kan sporadische uitgaven in geven
- Ziet overzicht krijgen van al zijn/haar inkomsten/uitgaven in chronologische volgorde
- Kan herhaaldelijke inkomsten in geven
- Kan herhaaldelijke uitgaven in geven
- Kan wijzigingen aanbrengen aan elk type inkomst of uitgave
- Kan zijn uitgaven/inkomsten filteren
- Kan zijn inkomsten en uitgaven exporteren
- Kan zijn inkomsten en uitgaven importeren
- Kan zijn inkomsten en uitgaven in categorieën opdelen
- Kan zelf categorieën creëren
- Kan meerdere rekeningen bijhouden bv.:
    - Cashflow in portefeuille
    - Zichtrekening
    - Spaarrekening
    - …
- …

STAPPENPLAN

Met behulp van het stappenplan gaan we stap voor stap de applicatie vorm geven. Na elke stap zou de applicatie steeds meer vorm moeten krijgen wat uiteindelijk resulteert in het eindresultaat. Echter dienen deze stappen als leidraad en hoeven deze niet verplicht gevolgd te worden. De tussentijdse stappen worden niet beoordeeld, een goed eindresultaat is des te belangrijker. Indien een stap dus niet direct lukt aarzel niet om een volgende stap op te nemen.

STAP 1

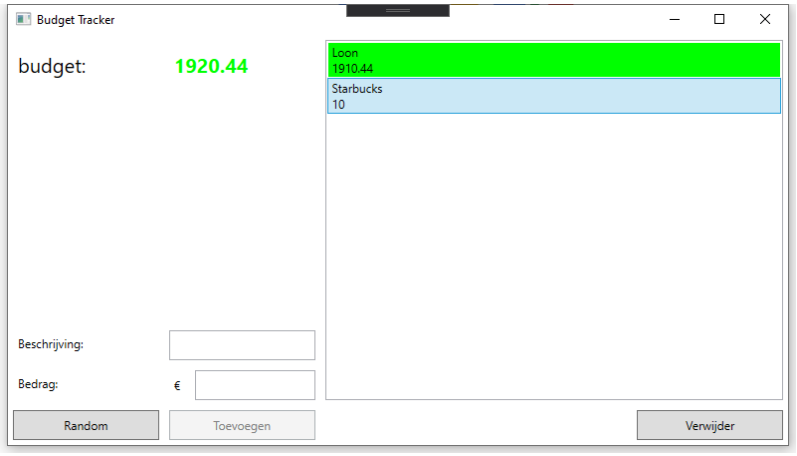
Creëer een basis GUI voor de applicatie. Deze bevat een listbox voor alle inkomsten en uitgaven, een label voor het huidige budget, een textbox om een beschrijving van een kost of uitgave in te geven, een textbox om een bedrag in te geven, een button om een random bedrag in te vullen, een button om een uitgave of inkomst in te geven en bijhorende labels.



1. Zorg ervoor dat het niet mogelijk is niet numerieke bedragen in te vullen of toe te voegen
2. Kleur negatieve bedragen rood en positieve bedragen groen
3. De random button voelt een random bedrag in.
4. Het budget wordt telkens aangepast op basis van de gegeven input.

Probeer de GUI op te bouwen aan de hand van een grid zodanig dat de lay-out behouden wordt als het scherm groter of kleiner wordt.

STAP 2



Maak het mogelijk om lijnen in het overzicht te verwijderen. De verwijder knop mag enkel functioneren wanneer er een item is geselecteerd. Na het verwijderen van een item moet het budget aangepast worden.

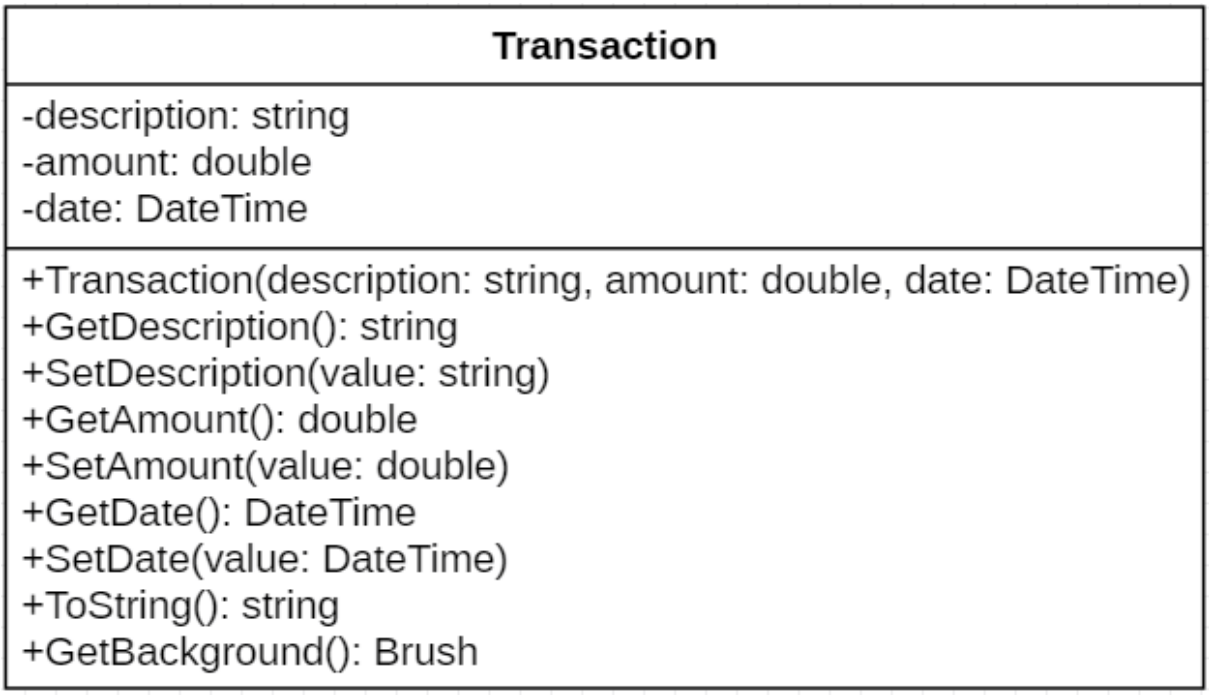
STAP 3

Maak een menu balk met 1 item “file” met daaronder 2 subitems “importeren” & “exporteren”.

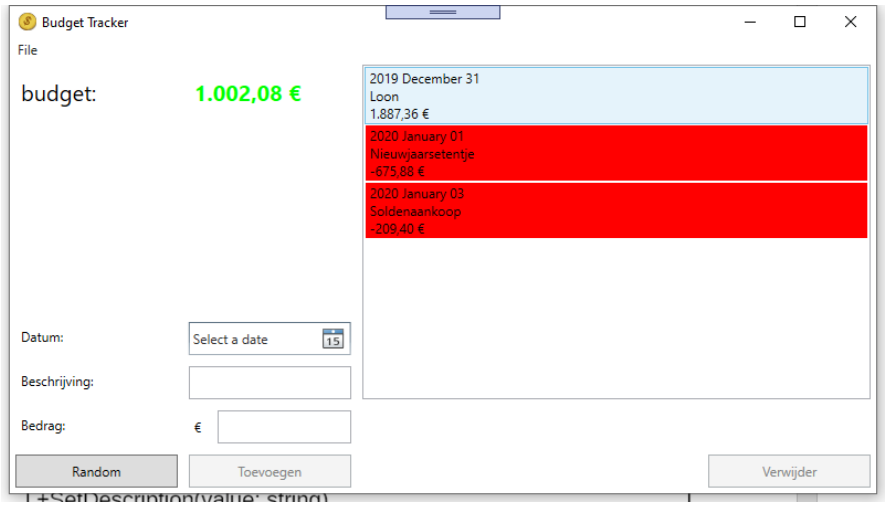
STAP 4

Zorg ervoor dat je de ingevulde data kan wegschrijven naar een bestand en dat je dat bestand terug kan inlezen. Dit kan je doen aan de hand van CSV files. CSV files zijn files die opgebouwd zijn aan de hand van een bepaald formaat. Wanneer je in excel een bestand maakt en dit opslaat als CSV, kan je wanneer je dit opent in een standaard texteditor ontdekken hoe dit bestand is opgebouwd.

STAP 5



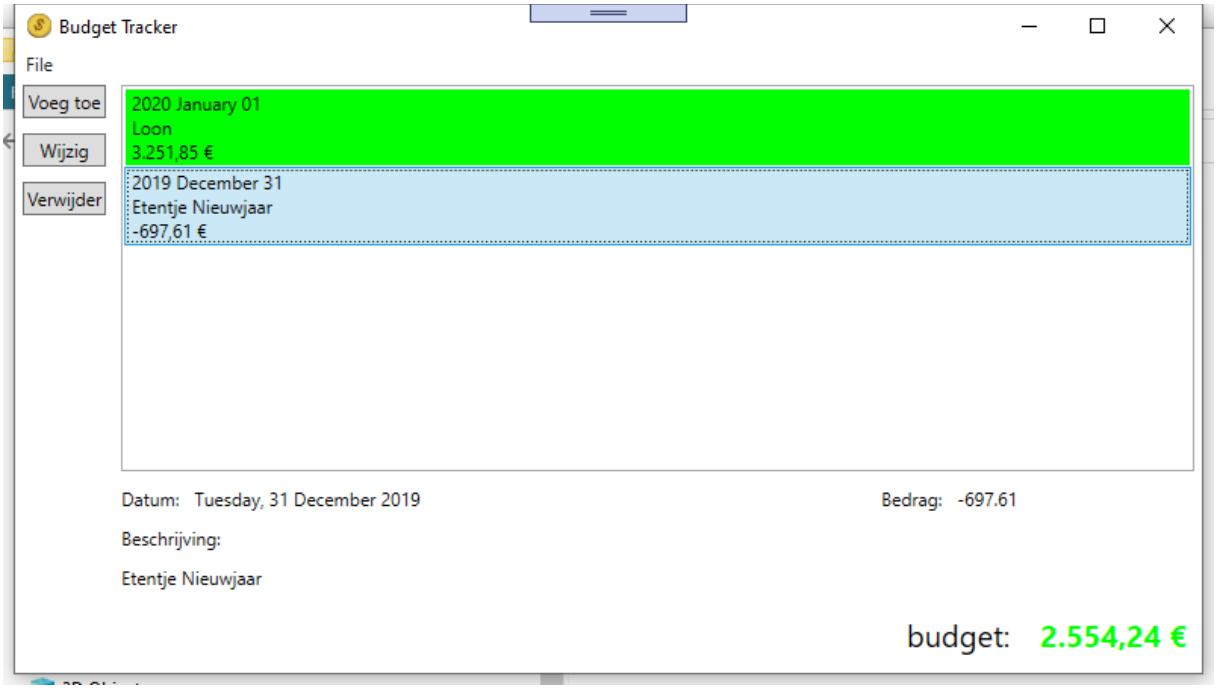
Breid de applicatie uit met een eigen gemaakte klasse. Deze vervangt de tot nu toe gebruikte items in de listbox. Voorzie in de GUI ook een veld om een datum in te geven, toon dit ook in het overzicht van inkomsten en uitgaven. Wanneer er geen datum ingegeven werd gebruik de datum van vandaag. Zorg er intussen ook voor dat het totaal budget berekend wordt op basis van de elementen in de listbox.



EXTRA: Wanneer je listItems volgens een bepaalde custom layout wil toevoegen kan je gebruik maken van een template en databinding. Dit valt buiten het bestek van deze cursus maar wie een uitdaging wil kan dit steeds proberen.

STAP 6

Voorzie in de applicatie een plaats waar je details kan zien wanneer je 1 van de velden aanklikt.

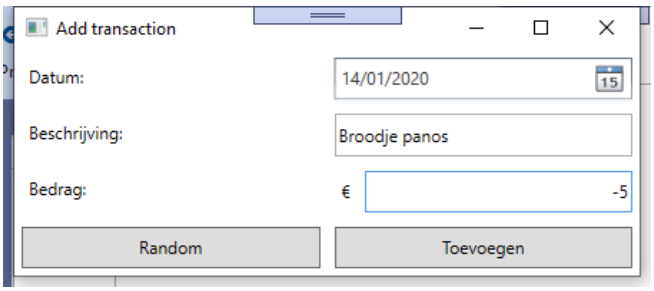


STAP 7

Voorzie een 2de scherm waar je data kan ingeven of wijzigen. Hiervoor kan je een 2de klasse die overerft van Window met bijhorende XAML file aanmaken. De klasse zou gemaakt kunnen worden op basis van onderstaand UML diagram.

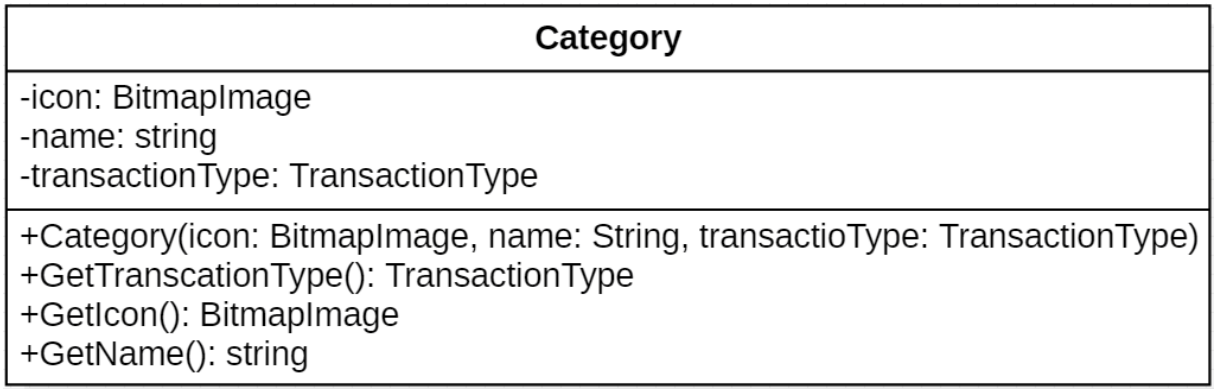
TransactionDialog
-transaction: Transaction
+TransactionDialog() +TransactionDialog(transaction: Transaction) +GetTransaction(): Transaction

Wanneer je in onderstaand scherm op toevoegen klikt, sluit het scherm en wordt de transaction toegevoegd aan de ListBox.



STAP 8

Zorg ervoor dat je aan transactions categorieën kan toevoegen. De klasse Category kan er uitzien zoals hieronder.

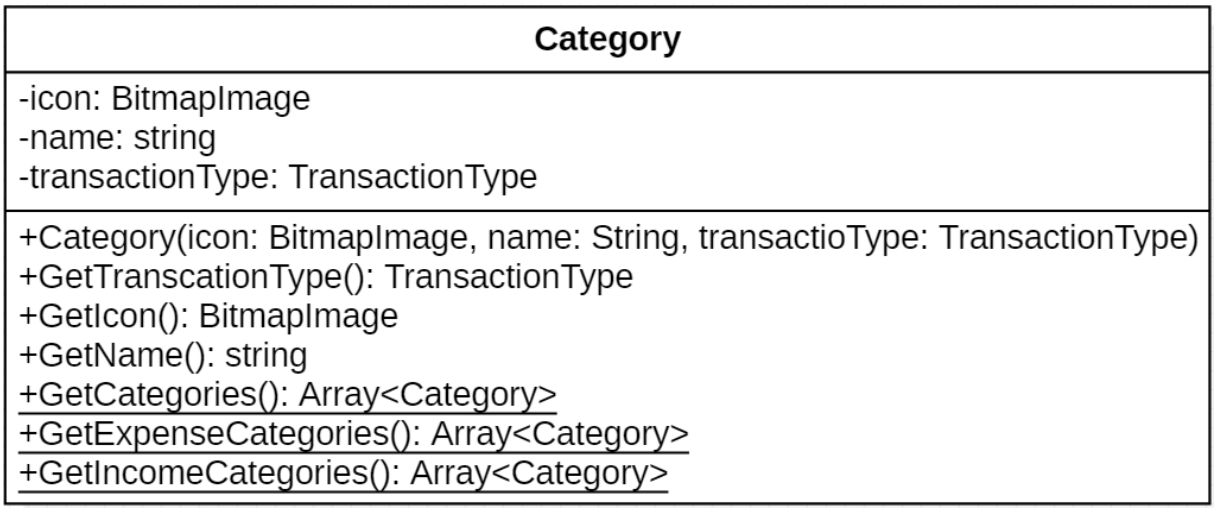


Merk op dat de klasse Category gebruik maakt van 2 andere “klasses”. Namelijk BitmapImage & TransactionType. De klasse BitmapImage is een bestaande klasse die voor ons voorzien wordt binnen de namespace `System.Windows.Media.Imaging`. De klasse TransactionType is een eigen gemaakte klasse en gaat eigenlijk een enumeratie zijn die bestaat uit 2 items namelijk Expense en Income.

Vergeet de klasse Transaction ook niet uit te breiden met een member category die de categorie van de transactie bijhoudt. Voorzie hiervoor ook de nodige extra aanpassingen binnen de klasse Transaction en je 2 view klasses.

STAP 9

Breidt de klasse Category uit met enkele static methodes die alle gemaakte categorieën bevat. De categorieën mag je hard-coded in de applicatie gebruiken. Gebruik deze methodes om in de GUI de hard-coded categorieën op te halen.



STAP 10

Zorg voor een 2de type transaction namelijk herhaaldelijke transaction. Dit zijn transactions met een startdatum einddatum en aanduiding of ze dagelijks, wekelijks maandelijks herhaald worden. Maak hierbij gebruik van overerving. Wees zelf creatief in hoe je dit implementeert in de GUI (TIP: met de methode GetType of het keyword “is” kan je veel doen)

UITBREIDING

In de requirements zijn er meer requirements genoteerd dan in de stappen doorlopen. Werk het project dus gerust verder af aan de hand van wat je momenteel kent. Of implementeer nog andere uitbreidingen die we niet in de requirements behandeld hebben.

PUNTENVERDELING

	Criteria	Score
Stijl (2)	Commentaar	1
	Stijlregels (naamgeving, hoofdletters,...)	1
Design (2)	Verzorgde & correcte XAML code	1
	Lay-out	1
Volledigheid (8)	Aanmaken transacties (met minstens datum, beschrijving en bedrag)	1
	Aanpassen transacties	1
	Opslaan transacties	1
	Inlezen transacties	1
	Verwijderen van transacties	1
	Toekennen van categorieën	1
	Aanmaken van herhaaldelijke transacties	1
	Controle van data (correcte meldingen wanneer nodig, enable/disable van GUI objecten)	1
Code (12)	Gebruik van correcte OO concepten (classes, enums, overerving, ….)	4
	Gebruik van correcte condities (selecties, iteraties)	3
	Correct gebruik van methodes (return types, ref vs value, parameters)	3
	Correct gebruik variabelen (declaraties, initialisaties)	2
Extra (6)	Per extra werkende uitbreiding kunnen er 2 punten gescoord worden (bvb.: Sorteren, filteren, zoeken, meerdere rekeningen, eigen categorieën maken, categorieën exporteren)	6
Minpunten	Compilatiefouten	-4
	Runtimefouten	-2
Totaal		**30**