

Custom Tkinter MVC テンプレート設計

pythonを使用したネイティブアプリの開発を効率的に行えるようテンプレートを作成。
Webフレームワークでよく使われるMVCアーキテクチャを採用。

要件

- CustomTkinterを利用したMVCテンプレートを作成すること。
- 画面は「メインページ」と「設定」の2つを用意する。
- 出来る限り、あらゆる実装方法が統一化できる構成にする。

MVCについて

システム開発においての設計方針の一つ。
このMVC構成は「Model」「View」「Controller」の3つに分けてコードを管理する構成。

役割

- Model: データ管理
- Controller: ビジネスロジック
- View: 画面表示

クラス種類 説明

No	クラス名	説明	備考
1	View	画面デザイン担当。	
2	Controller	ビジネスロジックを担当。ModelやViewの橋渡しやエラー処理などを行う。	
3	Model	データの保持を担当。	
4	CustomCtk	アプリ全体の設定 & コントローラの管理	コントローラの管理を任せることによりページ遷移などの共通処理を簡単に呼び出せる
5	App	各クラスの連携	CustomCtk、モデル、コントローラのインスタンス生成し、各クラスの連携を行う

命名ルール

1. 「Model」「Controller」に該当するクラスは語尾に必ず、該当する構成名を付ける事。

例: 「Modelクラス」 → OOModel
「Controllerクラス」 → OOContorller
上記のようにMVCに関するクラスどの担当をしているのかを分かるようにする。
※Viewに関しては名前をキーとして利用するため何のページか分かりやすいように命名する。

モジュール一覧

クラス一覧

※引数はコンストラクタの引数です。

No	クラス名	説明	引数	モジュール	備考
1	App	各種クラスの連携	引数1:self	app.py	
2	Page1	初期表示画面	引数1:self 引数 2:master(customCtk) 引数3:controller(インスタンス) 引数4:data(辞書型) 引数5:kwargs	page1.py	
3	Page2	2ページ目	引数1:self 引数 2:master(customCtk) 引数3:controller(インスタンス) 引数4:data(辞書型) 引数5:kwargs	page1.py	
4	BaseController	※抽象クラス	引数1:self 引数 2:root(customCtk) 引数3:ViewClass(ビュークラス) 引数 4:template_name(キ一名)	base_controller.py	BaseControllerを継承
5	Page1Controller	Page1画面のビジネスロジック	引数1:self 引数 2:root(customCtk) 引数3:ViewClass(ビュークラス) 引数4:my_model(モデルクラス)	page1_controller.py	

No	クラス名	説明	引数	モジュール	備考
6	Page2Controller	Page2画面のビジネスロジック	引数1:self 引数 2:root(customCtk) 引数3:ViewClass(ビュークラス) 引数4:my_model(モデルクラス)	page2_controller.py	BaseControllerを継承
7	MyModel	データ保持クラス	引数1:self	my_model.py	

イベント一覧

No	メソッド名	説明	引数	戻り値	使用メソッド	クラス名
1	goto_page	ページ2へ遷移 ※継承元のBaseControllerで定義したgoto_pageをそのままイベントへ登録する	引数1:self 引数 2:"Page2"	なし		Page1Controller
2	goto_page	ページ1へ遷移 ※継承元のBaseControllerで定義したgoto_pageをそのままイベントへ登録する	引数1:self 引数 2:"Page1"	なし		Page2Controller

メソッド・関数一覧

coreパッケージ

アプリ全体を管理する汎用的なコードを保持。

CustomCTk

BaseController

App

modelsパッケージ

データ管理を担当。

mymodel.py

controllersパッケージ

ページごとのコントローラを保持。

page1_controller.py

page2_controller.py

viewsパッケージ

ページごとのビューを保持。

page1.py

page2.py

メモ

View

- Main_View
- Settings_View

Controller

- Main_Controller
- Settings_Controller

Model

- Settings_Model