WeightTracker.javaについて

# コード解説

import android.content.Context;

* Androidアプリ内の情報を得るための **Context** を使うためのimport
  + Contextとは？
    - 今の「**アプリや画面の状態・環境**」の情報をまとめて持ってるオブジェクト

グラフィカル ユーザー インターフェイス

AI 生成コンテンツは誤りを含む可能性があります。

public WeightTracker(Context context) {  
 this.context = context;  
}

* このクラスのコンストラクタ
  + アプリの環境（Context）を渡して、それを使えるようにクラスに保存している
  + このthisは**activity**のcontextを渡している
* Room.*databaseBuilder* (): Roomのヘルパー関数で、データベースを作るための **ビルダー（Builder）オブジェクト** を返す

public void someMethod() {  
 *// これで getApplicationContext が使える！* Room.*databaseBuilder*(context.getApplicationContext(), SaveRoom.class, "your-db-name").build();  
}

* + ビルダーオブジェクトとは？
    - オブジェクトを段階的に組み立てるための設計パターン（Builderパターン）で使われる

テキスト, 手紙

AI 生成コンテンツは誤りを含む可能性があります。

テキスト, 手紙

AI 生成コンテンツは誤りを含む可能性があります。

* context.getApplicationContext()

テキスト, メール

AI 生成コンテンツは誤りを含む可能性があります。

※Activityのcontextを間違えて使ってしまうと、**画面が消えているのに、そのcontextを使おう**とする(長く使うものじゃない)

→　アプリがクラッシュしたりメモリリークする

この**クラッシュ等を防ぐため**にcontext.getApplicationContext()を使用して、「長く使うcontext」に変換している。

* SaveRoom.class
  + @RoomDatabaseクラスを継承したクラス
  + デーブルやDaoを定義している

なぜ引数で渡す必要があるのか

こういうテーブルがあるよなど、データベースの設計図を渡すため

テキスト, 手紙

AI 生成コンテンツは誤りを含む可能性があります。

* your-db-name
  + データベースのファイル名。アプリの内部ストレージに " your-db-name " という名前で保存される。
* .allowMainThreadQueries()
  + デフォルトでは、Roomはメインスレッド（UIスレッド）でのDBアクセスを禁止してる。重くてUIが固まる可能性があるから。
  + それを強制的に許可するメソッド
  + 使用は**推奨されていない**らしい
    - 操作がカクカクになるから
* build()
  + 最後に .build() を呼ぶことで、**RoomDatabase のインスタンスが完成**する。
  + このインスタンスを変数に保持しておくことで、DAO（Data Access Object）を使ってデータベース操作ができるようになる。

double previousWeight = readPreviousWeight(); // 前回の体重を取得

* データベースから最新の体重を読み取って変数に保存する

if (previousWeight != -1) {

double difference = currentWeight - previousWeight;

System.out.printf("前回との差: %.2f kg\n", difference);

} else {

System.out.println("前回の記録がありません。");

}

テーブル

AI 生成コンテンツは誤りを含む可能性があります。

SaveRoom db = Room.databaseBuilder(

context.getApplicationContext(),

SaveRoom.class, "weight-database"

).allowMainThreadQueries().build();

* RoomDatabaseを初期化している