すべての設計が完了したので、実際に機能を記述する準備ができました。

冒頭で述べたように、デザインはこのアプリの最も難しい部分であり、実際にBMIを実際に計算する機能は最も簡単です。

BMIを計算する機能を含む新しいDartファイルを作成することから始めましょう。

多分それをcalculator\_brainと呼びましょう。このファイルでは、マテリアルコンポーネントは必要ありませんが、作成するのは新しいクラスです。

そして、CalculatorBrainと呼ばれます。そして、このクラスには2つのプロパティがあります。

整数になる高さを渡します。また、重みを渡します。これらのプロパティを渡す方法は、もちろん電卓の頭脳を構築するときです。

それでは、身長と体重を持つコンストラクターを作成しましょう。

コンストラクターができたので、電卓の脳から新しいオブジェクトを作成するときに、身長と体重の値を提供できるようになります。そして、いくつかの機能の記述を開始できます。

最初に必要なことは、実際のBMIを計算できるようにすることです。

現在、BMIはボディマス指数の略です。そして、それは身長に対して誰かの体重を正規化する方法です。

したがって、BMIの計算式は、キログラム単位の質量または重量をメートル単位の高さで除算した値です。

しかし、現在の身長はセンチメートルであるため、ここで少し調整する必要があります。

先に進んで、BMIを文字列として返す関数を作成し、calculateBMIと呼ぶことにします。計算機の脳が初期化されると、身長と体重が必要なものすべてに実際にアクセスできるようになるため、これは入力を取りません。

しかし、それはいくつかの仕事をするつもりであり、それがやろうとしている仕事は、倍になるBMIを計算することです。そして、BMIを記録し、すでにキログラム単位の高さを二乗した高さで割った値に等しくなります。

したがって、前に見たように、数字を2乗する最も簡単な方法は、Dart数学ライブラリを使用することです。

数値と指数を取るべき関数を使用できるように、それをインポートしましょう。したがって、高さを提供することができ、次に、この場合、高さの2乗までの力を提供することができます。

さて、私たちの身長はセンチメートルであると先ほど言ったことを思い出してください。

したがって、メートルに変換するには、100で割る必要があります。

これで、この非常に単純な式からBMIが計算されました。これを使用して、文字列に変換できます。

したがって、この時点で、BMIはダブルだと言ったので、非常に長い値になります。これで、アプリに必要なのは、実際には小数点以下1桁だけです。

それで、18.100万行の長さではありませんが、それは非常に大きいので、画面から消えてしまい、誰もその程度の精度を必要としません。 doubleを小数点以下1桁に変換し、文字列に変換するには、本当に便利な方法があります。したがって、BMI.toStringAsFixedを記述すると、小数点文字列が返され、入力として小数点以下の桁数を指定できます。

そのメソッドを使用して、小数点以下1桁だけが必要で、returnキーワードを使用してこの値を出力できるとしましょう。

したがって、calculateBMIは体重と身長に基づいてBMIを計算できるようになり、単一の小数点値に変換して文字列として返すこともできます。

次に必要なのは、そのBMIに基づいて結果を提供できるようにすることです。

したがって、結果となる文字列を返す別のメソッドを作成し、メソッドgetResultを呼び出します。

この場合も入力は必要ありませんが、BMIが25以上であるかどうかをこのメソッド内でチェックします。この場合、Overweightという単語が返される可能性があります。

そして、25を超えていないかどうかを確認します。18.5を超えている可能性があります。

その場合、結果は正常になります。

最後に、18.5よりも低いすべての結果について、単純に戻ります。

体重不足。

現在、これらの解釈は、ほとんどの一般的なBMIチャートに基づいています。

必要に応じて、ウィキペディアまたは他の場所。

そのため、現時点ではBMIにアクセスできませんが、このエラーが発生するのはこのためです。

ここで計算されるBMIはこれらの中括弧に限定されているため、BMIについては知りません。

ここでローカルにのみ表示されます。

しかし、アクセスしたい場合は、このメソッドから見えるようにする必要があります。

それでは、ここまでプライベートプロパティを作成しましょう。

したがって、ダブルとして作成し、プライベート変数になります。そのため、事前にアンダースコアを追加しますが、最初は何もありません。

そして、BMIを計算するときに、その値にBMIを割り当て、それを文字列として返すこともできます。

そして、そのBMIが必要なすべての場所で使用できるようになりました。ただし、これはプライベートな値であるため、BMIにアクセスしようとしたり変更しようとしたりするビジネスはないので、電卓の脳以外のクラスからアクセスすることはできません。必要な場合は、このメソッドを介して取得できます。このメソッドは、必要な方法で返すだけで、1つの小数点です。しかし、CalculatorBrain内では、これに自由にアクセスでき、結果を計算するために使用できますが、解釈に使用することもできます。

そこで、getInterpretationという別のメソッドを作成してみましょう。これにより、同じ3つのチェックが実行されます。

ここにコピーして貼り付けます。

ただし、通常の過体重、過体重を返すのではなく、体重の意味についてユーザーに少しフィードバックを提供できるようにしたいと考えています。そして、あなたが本当にしたいなら、多分少しのアドバイス。私が1時間入力するのを防ぐために、先ほど書いたいくつかの解釈を単にコピーして貼り付けます。

これらを使用するか、独自に作成することができます。それはあなたのアプリですので、あなたのやり方でやってください。しかし今、私たちは電卓の脳内に3つのメソッドを持っています.1つはBMIを提供し、1つはBMIの結果を提供し、1つはBMIの解釈を提供します。

これで準備が整い、CalculatorBrainクラスを使用する準備が整いました。

それでは、先に進み、計算機の頭脳を現在のファイルにインポートしてみましょう。完了したら、これらのインポートをすべて折りたたむことができます。そして、それが必要なのは、ユーザーが計算ボタンをクリックした瞬間です。

そのため、onTap内では、新しい計算機ブレインオブジェクトを初期化することになります。そして、私はそれを短くするために単にcalcと呼ぶつもりです。そして、前に構築したコンストラクターを使用して初期化されます。

したがって、ここでスライダーから得た高さと、入力ページからも得た重さの高さを期待します。

それでは、重みをローカルの重みに等しく設定しましょう。

計算機ブレインを作成し、BMIの計算を取得し、解釈を取得し、結果を取得できるようになったので、それを結果ページである次のページに渡すことができるようにします。

したがって、結果ページに移動すると、現時点では外部からアクセスできるプロパティがありません。

ただし、ここをナビゲートするときにデータを渡すために、これらのプロパティのいくつかを作成する必要があります。

それでは、bmiResultと呼ばれる最終プロパティを作成してみましょう。これは文字列になります。

そして、resultTextと呼ばれる別のものを作成しましょう。

そして最後に、解釈と呼ばれるものがあります。また、初期化が必要な最終プロパティがあります。

それでは、結果ページコンストラクターを作成しましょう。結果ページを読み込み、変に見えないようにするために、率直に言って、すべての結果を必要なものとしてマークします。結果テキストと解釈。

したがって、3つすべてを必要に応じてマークしましょう。

this.interpretation、this.bmiResult、this.resultTextがあります。

そのため、ナビゲートしたい瞬間に結果ページを作成するときに、電卓の脳から取得した値を渡すことができます。

したがって、初期化できる3つのプロパティがあります。

1つはBMI結果、もう1つは結果テキストであり、最終的に解釈が得られました。

したがって、BMIの結果は、BMIを計算するための計算機ブレインを使用したcalcと等しくなります。

そして、これは、ここでそのプロパティBMI結果に直行する文字列を返し、ビルドメソッド内で使用できるようになることを忘れないでください。

これで、次のメソッドはgetResultメソッドを使用して計算され、最後のメソッドはgetInterpretationメソッドに依存します。

これで、電卓の頭脳からすべての文字列を取得できたので、ビルドメソッドで使用できます。

したがって、これらのハードコードされたテキストを使用する代わりに、結果テキスト、BMI結果、および解釈を実際に追加できます。

ここでスタイルと一貫性を保つために最後に行うことは、ここで結果テキストのすべての単語を大文字にすることです。そしていつものように、これを行う簡単な方法が常にあります。

そして、この場合の1つはtoupperCaseと呼ばれ、これは単に文字列内のすべての文字を大文字に変換します。これが必要なものです。ここで、アプリを停止し、最初から起動するようにゼロから実行します。

それでは、BMI計算機をテストしてみましょう。

私たちはそれをコーディングすることから、またウィキペディアから、性別は本当に重要でないことを知っています。しかし、アプリの別の部分で使用している可能性があるため、アジェンダを選択してカードを有効化できます。また、身長が180になるように身長を選択して、体重を調整し、年齢を調整して、計算をクリックします。

したがって、BMIが計算され、正常であることがわかり、BMIの解釈がわかります。私のBMIは小数点以下1桁に丸められ、この巨大な数字として表示されます。

ここで、もう少し低い高さ、少し大きい重量でもう一度試してみたい場合は、計算をクリックしてください。異なる結果と異なる解釈が得られることがわかります。

このモジュールでは、多くのことを学びました。特にDartプログラミングを初めて使用する場合は、これらの重要な概念のいくつかを確認する価値があります。しかし、この美しいBMIアプリを私と一緒に構築するのが楽しいことを願っています。また、UIデザイン用にウィジェットをカスタマイズするさまざまな方法を使用して、Flutterを次のレベルのデザインに導いてください。

このモジュールでは、それがすべてです。

私は次のものを見る。