前回のレッスンでは、Dartを使用してクラスとオブジェクトを作成する方法を見てきました。

しかし、なぜ実際にそれらが必要なのでしょうか？

彼らはどのような目的を果たしますか？

これを理解するには、まずコンピューターが実際にどのように機能するかを理解する必要があります。金属製のシェルときれいなキーボードと素敵な画面の背後にあるコンピューターは、多数のスイッチで構成されています。そして、これらのスイッチは、オンまたはオフの2つの状態になります。そして、多くの異なるスイッチをオン/オフすることで、コンピュータに何をすべきかを伝えることになります。

コンピューターをプログラムするためにそこに座ってこれらのゼロをすべて書かなければならなかった場合、何年もそこにいることになります。

しかし、その後、文字Aを表すために、

01000001と書かなければなりません。それは非常に苦痛で、非常に時間がかかります。

これは、マシンコードと言います。

そのため、英語のように聞こえるプログラムを簡単に作成するためのあらゆる種類の方法を考え出しました。そのため、コンピューターレベルに降りて1と0を書き始める必要なく、実際にコンピューターと対話できます。

初期の頃は、多くのプログラミング言語が手続き型と呼ばれていました。

彼らは上から下に行くだろう/だからこれをし、それからこれ、それからこれ、そしてこれから、基本的に永遠に。

そして、これはちょっとわかりやすいです。しかし、コードがますます複雑になると、やや毛深い原因になり、何かがうまくいかなかったとき、正確に何が間違ったのかを知るのが難しくなりますか？

上から下までロジックに従う必要があり、それを解く必要があり、本当に簡単に自分の足で撃つことができるからです。

多くの場所で、複雑さは常に敵です。

人間としては非常に有能な脳がありますが、一度に保持できる情報は限られています。プログラムが複雑になるほど、プログラムがクラッシュし、問題が発生し、パフォーマンスが低下する可能性が高くなります。

オブジェクト指向プログラミングのこのコンセプトが登場し、多くの人々がこのコンセプトを本当に愛していました。

ですから、オラクル、Java、Apple Swift、実際に私たちが取り組んでいるDartなど、本当に大きくなった言語はすべてオブジェクト指向プログラミング言語です。

それはどういう意味ですか？

それは、アプリやプログラムで必要なことをすべて行うためにオブジェクトを操作していることを意味します。

また、ソフトウェアエンジニアとしての仕事を探している人にとって、最もよく聞かれる質問の1つは、「オブジェクト指向プログラミングの4つの柱は何ですか？」です。

オブジェクト指向プログラミングとは何ですか？そして、人々が巻き戻すことを非常によく耳にする4つの柱は、抽象化、カプセル化、継承、および多態性です。

今では一口ですが、これらの大きな言葉が実際に何を意味するのかを完全に理解している人はほとんどいません。

そこで、私はそれを分解して、アプリのコンテキスト、OOPがなぜ素晴らしいのか、そしてそれがDartにどのように適用されるのかを説明したいと思います。

最初の柱の1つは抽象化です。

複雑なものをより小さな部分に分割して、より抽象的なものにするにはどうすればよいですか？

今までに私のコースを受講したことがあるなら、あなたは私がどれだけ食べ物が好きかを知っています。

したがって、抽象化を理解するために、ご想像のとおり、食べ物について説明します。

日本で私のお気に入りのレストランの1つは、4〜5席しかない非常に小さな穴のあいた場所です。しかし、ウェイター、シェフ、清掃スタッフである一人の男もいて、バーカウンターの後ろからすべてをやります。注文するときは、自動販売機で注文し、彼に注文します。そして、彼はあなたに注文を料理します、彼はあなたに代わってあなたにテーブルに置きます。

そして、あなたが完了したら、彼はあなたのボウルを取り、彼はそれを食器洗い機に入れます。

今、これは本当にクールですが、いくつかの制限があります。彼は10人以上の顧客を言うことができないでしょうか？

すべての仕事を自分でやるのは完全に悪夢だからです。

そして、それは一種の欠点です。

多くのことをしようとしている大きなコードがある場合、料理をしている従業員、スタッフを待っている人、レジで支払いをしている人、予約をしている人、予約。そして、まず、CVを使用してこれらのすべてを実行できる人を見つけることができますように。

そして、あなたがそれらを雇うならば、あなたは1人がそれほど多くのことをする必要がないので、あなたはそれらを完全に破壊しようとしています。

そうした場合、サービスを提供できる顧客の数、レストランの大きさ、アプリやプログラムの複雑さなどに制限が生じます。

あなたのプログラムがより複雑でより面白くなるにつれて、あなたがしたいことはモジュール式の上昇が欲しいということです。

より抽象化し、異なるジョブを分離したい場合。そのため、実際に食べ物をお客様に届けるウェイターがいるでしょう。

ケーキを作るパティシエがいます。スーシェフと残りの料理をするシェフがいます。

そして今、それは誰もが自分の仕事を持っているということです。

彼らは皆、自分が何をする必要があるかを知っており、お互いの道に入り込まず、一度にすべてをやりすぎる必要はありません。

そして今、あなたはあなたのレストランを拡大することができます。より多くの人々に奉仕することができ、より抽象的にもっと面白いことをすることができます。

したがって、プログラムを作成している場合、すべてを知っているすべてのことをすべて実行できる単一の大きなコンポーネントを作成するのではなく、その種はすべてが混ざり合っているスパゲッティコードと呼ばれるものになりがちです。

そして、何かがうまくいかなかったり、期待したとおりに動作しないときにデバッグしようとするのは面倒です。

代わりに、機能を異なるコンポーネント（シェフコンポーネント、ウェイターコンポーネント、受付係、パティシエシェフ）に分割することをお勧めします。

そして、これが抽象化のすべてであり、定義されたジョブまたは定義されたロールを持つより小さな部分を作成することでより複雑なシステムを構築し、実際に複雑さを管理できます。

コードに適用しようとすると、実際にはどういう意味ですか？

現時点では、main.dartを見ると、かなり長くなっていることがわかりますか？

そして、多くのことをしようとしています。

たとえば、質問バンクの質問があり、新しい質問を取得する必要があるときに質問バンクを調べる必要があるときに、現在どの質問にあるかを追跡しようとします。

また、アプリのレイアウトを配置するなどのことも行います。ボタンとそのすべての機能を実装できることはご存じのとおりです。そして、ここは少し面倒です。

アプリがさらに複雑になった場合、このmain.dartファイルで何が行われるかについて、少し混乱するかもしれません。

それでは抽象化しましょう。よりモジュール化してみましょう。それを行うために、libフォルダーに移動します。右クリックして、新しいDartファイルを作成します。

これがクイズの頭脳になるため、このDartファイルをquiz\_brainと呼びます。

Gitに追加するかどうかを尋ねられたら、先に進み、クリックします

はい。

これで、新しいquiz\_brainを作成できます。もちろん、クラスになります。

そこで、Quiz brainというクラスを作成します。クラスは大文字で始めなければならないことを忘れないでください。

そして、このクラスでは、クイズがすべきこと、そしてできるべきことをすべて定義します。

まず、クイズにはいくつかの質問が必要です。

main.dartに進んで、質問バンクを他のすべての機能と混合する代わりに、ここから切り取り、quiz\_brain内のあるべき場所に配置しましょう。

最初に表示されるのは、ここにいくつかのエラーがあることです。

この質問の内容がわからない。

それは、そのクラスの場所であるquestion.dartファイルをまだインポートしていないためです

質問は現在定義されています。

質問ファイルのトピックについて説明しますが、最初に注意することは、現時点では、このクラスは非常に単純な仕事をしているということです。

それはすべての質問の構造を定義します。つまり、質問があり、答えがなければなりません。質問クラスのコンストラクタを作成するとき、ここでパラメータに名前を付けました。

したがって、新しい質問オブジェクトを作成するには、質問newQuestionを作成する必要があります。

そして、qとaも指定する必要があります。

これは少し冗長です。すべての質問に文字列の一部であるテキストとブール値である回答があるだけだとすると、

文字列を含む質問を作成した場合、それを確認できるのは非常に簡単です。

これはテキストであり、ブール値を持っています。これが答えです。

そこにパラメーター名qとaを持つ必要はありませんが、これもあなたの判断です。

そのため、他の方法を好む場合は、他の方法で保管してください。

しかし、質問の大規模なバンクを構築し始めると、これははるかに読みやすくなります。

質問の大規模なバンクといえば、それは3つの質問だけでクイズアプリの多くではないですか？

そのため、readmeファイルを開くと、事前に入力された質問が既に含まれていることに気付くでしょう。

それらの13があります。

そして、バックティックの間のすべて、それらの質問すべてをコピーすると、それを貼り付けて、現在の質問バンクにあるものを置き換えることができます。そして、それらはすでにフォーマットされているため、質問テキストと回答を含む新しい質問オブジェクトを作成します。今、私たちがしなければならないのは、メインに行くことです

.dartファイルを作成し、ここにあるエラーを修正します。質問バンクをメインから削除したため

.dartファイル、これはもはや存在せず、それだけであなたに伝えます。しかし、どうすればquiz\_brainを参照できますか？

さて、新しいクイズ脳オブジェクトを作成する必要があります。最初にquiz\_brainをインポートする必要があります

.dart、それをそこに入れましょう。

そして今、quizBrainと呼ばれる新しいクイズ脳オブジェクトを作成することができます。

そして、オブジェクトに名前を付けるとき、それらは小文字で始まり、新しいクイズ脳オブジェクトに等しくなるように設定することに注意してください。

これで、quizBrainを参照することで利用できる新しいオブジェクトが作成されます。ここに質問バンクがある場所では、ローカルに存在する質問バンクだけを使用する代わりに、quizBrain.questionBankを作成します。これは、質問バンクがクイズ脳オブジェクトのプロパティになったためです。そしてここにあります。質問のリストです。これで、main.dartファイルの質問について知る必要がなくなり、この未使用のインポートを削除できます。インポートでコードを要求するのをやめるたびに、それを灰色にして波線を付けることで通知することに注意してください。

それを削除しましょう。main.dart内のクイズブレーンについてのみ知る必要があります。

ここにまだいくつかのエラーがあります。

ですから、あなたへの私の挑戦は、すべてを修正して実行し、同じように動作することを確認することです。

ビデオを一時停止し、それを試してみてください。大丈夫。

彼女の下で、前とまったく同じことをします。質問バンクはクイズの頭脳の財産であるため、代わりにドット表記を使用する必要があります。

quizBrain.questionBankという書き込みがあります。ここでも同様に、quizBrainを作成する必要があります。

questionBank。実際の質問バンクを実際に利用しています。今、main.dartにもう少しコードを追加する必要がありましたが、少なくとも今では非常に大規模な質問バンクは、別のクラスに分離されています。そして、これにより、クイズの頭脳などの機能の一部が別のオブジェクトに抽象化されます。

そして今、コードをもう少し抽象化しました。もう1つの利点は、別のクイズを作成する場合、たとえばスポーツクイズを作成する場合、main.dartについて何も変更する必要がないということです。

スポーツクイズの頭脳と言う新しいファイルを提供するだけで、ここにあるものはすべて同じように機能します。

そして、クイズ脳を構築するために使用しているクラスを変更する必要があります。

したがって、さまざまな種類のクイズ脳、スポーツ、ニュース、一般的な知識を備えたクイズを持つことができ、一方を他方に切り替えるたびに、現在のクイズに実際に使用しているオブジェクトを変更するだけです。

そのため、コードベースの再利用性とモジュール性を高め、さまざまなジョブをさまざまなクラスに分離しました。

ここで、アプリを実行すると、13個の質問が追加されていることがわかります。

しかし、ここに問題があります。私たちのクイズアプリでは、負けるのが本当に嫌いだとしましょう。

質問に対する答えを実際に変更できます。

したがって、たとえばここでは、クイズの頭脳をタップし、質問バンクのプロパティを見て、質問バンクから現在の質問を取り出して、答えをタップすることで正しい答えを取得しようとしています。

そして、これを取得と呼びます。私たちはその正解の価値を得ています。

ただし、同様に変更することもできます。

クイズ脳の質問バンクの現在の回答を書くだけで、新しい値を設定できます。選んだものは何でも設定できますが、この場合は正しいでしょうか？

私の答えを確認する前にこれが発生した場合、保存をクリックして実際の正しい答えと結果を見てみると、この質問で私たちが生み出した最も大きな音についていることがわかります動物と正解は偽である必要があります。

しかし、私はとにかく真実を押すつもりであり、それは私がそれを正しくしたことを教えてくれます。

それではどうして可能ですか？常に正しいことをしたい場合でも、それを変更することはできません。

これはアプリにとって非常に安全ではありません。

それについてコメントしましょう。次のレッスンでは、コードを少し改善し、OOPの2番目の柱であるカプセル化に準拠させる方法を学習します。

それで、そのすべてについて、次のレッスンでお会いしましょう。