大丈夫。

クラスとオブジェクトの動作を確認し、それらを少し使用したので、Dartでクラスとオブジェクトがどのように機能するかを詳しく見ていきましょう。

まず、クラスは単なる青写真であり、アプリで表示するすべてのものを作成するために使用され、ユーザーは対話します。したがって、本質的に、アプリを構成するすべてのものは、クラスと呼ばれる設計図を使用して作成されます。

では、オブジェクトとは一体何なのでしょうか？

その青写真を撮って工場に持っていくと、実際の車はこの例では構築されます。それがオブジェクトです。それが運転を行う実際のことです。アプリの場合、タップを受け取ることができる実際のボタンと実際に変化する画像になります。

これらは実際のオブジェクトです。しかし、アプリが実行されるまでオブジェクトを見ることができないため、これらのクラスを作成して、それらがどのように機能し、どのように機能するかを計画する必要があります。

クラスには2つの非常に重要なことがあります。 1つはそのプロパティです。

ですから、私たちが車を作り上げているのなら、それはその色、持っている座席の数、持っている車輪の数、持っているドアの数、またはその特性に関係するあらゆる種類のものになりますオブジェクト。そして、Dartでは、これらがクラス内にあるときに変数と呼んでいたものが、プロパティまたはフィールドと呼ばれるようになりました。

ここで、定義する必要があるクラスの他の重要な部分はメソッドです。

それができること。最終的に作成されるオブジェクトが実行するアクションが実行可能になります。

私たちの自動車オブジェクトの場合、それは運転すること、ブレーキをかけること、または向きを変えることです。

そして、これらのメソッドはオブジェクトが実行できるアクションであり、オブジェクトが構築される前であっても青写真で定義します。

たとえば、このように定義されたクラスが表示される場合があります。

まず、クラスキーワードがあります。これは、後から来るすべてが青写真になることを示しています。

そして、クラスの名前があります。

この場合、車の設計図を作成しています。そして、他の多くの言語と同様に、Dartの慣例により、変数を作成する場合はここにnumberOfDoorsがあり、関数を作成する場合はここにあります。小文字の単語。

しかし、クラスが異なることを示すためにクラスを作成する場合、常に大文字から始めます。

したがって、クラス名のすべての単語は大文字になります。

これがCarクラスで、これらの中括弧はこのcarクラスの定義です。

これが、私たちの車がどのように見えるか、またそれがどのように振る舞うかの正確な青写真です。

そして、ルックスと動作の間で、クラスになります。

したがって、この場合、numberOfDoorsと呼ばれるプロパティが1つだけあります。

そして、これは車の特性、つまり、ドアの数を定義します。

この車には5台あります。

この車にはドライブと呼ばれる機能もあり、これは車の機能を少し定義します。

つまり、この車は、ドライブと呼ばれるものを呼び出すと、たとえば車輪を回し始めます。

ここで、クラス内で変数が変数と呼ばれなくなり、プロパティまたはフィールドと呼ばれることに注目してください。

同様に、クラスの中括弧内で定義された関数は、関数ではなくメソッドと呼ばれるようになりました。

プログラミングに慣れていない場合は、新しい単語がたくさんあります。新しい言語を学習するようなものなので、独自の単語の用語集を作成することをお勧めします。

人々がメソッドを参照するとき、それらはクラスに関連付けられた単なる関数と考えることができます。

そして、人々がプロパティについて話しているとき、それは単にクラスに関連付けられた変数です。

そして、それを十分に頻繁に聞くと、それは第二の性質になり始めます。

オブジェクトを取得したら、移動などのクラスで事前に計画されたアクションを実際に実行したのはオブジェクトですか？

または、実際にクラスで定義した特性（colorプロパティやnumberOfDoorsプロパティなど）があります。オブジェクトを作成する2つのパスが最初であり、クラスを作成する必要があります。これは、作成される最終的なオブジェクトに関するすべてを定義する青写真になります。

そこで、classキーワードを使用し、クラスに大文字の名前を付け、そのクラスのプロパティである変数を定義します。

また、クラスが実行できるメソッドになる関数も提供します。

そして、実際にこのクラスからオブジェクトを作成すると、コードは次のようになります。

型（この場合はクラスの名前）を定義し、オブジェクトに名前を付けます。

そして最後に、そのクラスの新しいバージョンに等しくなるように設定します。 carという単語とこれらの空の括弧がある場合、基本的にこの青写真から新しいオブジェクトを作成しています。

その青写真を工場に送り、実際の車のオブジェクトが出てくるようなものです。

今、これは非常に抽象的なものであり、頭の中でその多くを想像する必要があります。

それをコードに書き留めて、信頼できるDartPadに向かい、これを実際に見て少し理解を深めましょう。

ここに空のメインメソッドがあり、そのすぐ下に新しいクラスを作成します。

そのクラスをHumanと呼びます。

したがって、大文字にする必要があることを忘れないでください。

そして、このクラスの人間にはいくつかの特性があるため、いくつかの特性があります。

それでは、人間にはどのような特徴があるのでしょうか？

まあ、彼らは正しいでしょうか？

heightプロパティを作成でき、おそらくageプロパティがあります。そして、新しいクラスを作成するとき、これらのプロパティを空またはnullとして作成することができます。または、デフォルトで、新しい人間を作成するとき、常に高さを持つ必要があると言えます。

デフォルトの人間が作成されると、高さ15センチメートルである可能性がありますが、デフォルトの年齢が0にな​​る可能性があります。

メインメソッドに移動して、この人間から新しいオブジェクトを作成する場合、新しい人間を作成するとしましょう。これがこのオブジェクトのタイプになります。そして、私たちの実際の人間は、彼女のジェニーと呼びましょう。人間のクラスを使用して作成されます。そしてもちろん、人間を書くことでそれを行い、空の括弧のセットを用意します。

そして、これは私たちの新しい人間を構築します。

したがって、人間はすべての人間の青写真です。

彼らはすべて身長があり、年齢が必要です。

そして、人間が最初に構築されたときの人間の状態のデフォルト値がいくつかあります。

そして、ジェニーはこの青写真から構築された実際のオブジェクトです。そして、私たちはプログラムで物事を行っています。実際には、人間のための抽象的な設計図のようなものではなく、ジェニーとやり取りしています。ジェニーの身長を印刷しましょう。したがって、printステートメントがあり、次にJennyオブジェクトを利用して、ドット表記を使用してそのプロパティ、つまりheightプロパティを利用します。

したがって、jenny.heightと言い、コンソールに出力しましょう。すべての人間にデフォルトの開始身長15を指定したため、ジェニーの身長も15になっていることがわかります。

少し後で、それを変更することができます。

多分ジェニーは少し成長して、彼女の身長が20cmになったと言いましょう。

したがって、これらのすべてのプロパティを変更し、ドット表記を使用するだけでそれらを利用できます。これは私が心配しているオブジェクトであり、人間のクラスから作成されたことがわかっているオブジェクトを見つけ、デフォルトでは、人間のすべてのプロパティはこれらの値で始まります。それから私はジェニーの身長を教えてください。そして、それはそのドットが言うことです。

「これは私が知りたいジェニーに関連する財産です」と言っているだけです。

または、後で人間がそれに関連付けられたいくつかの機能を取得すると、それを利用する方法はまだドット表記法であることがわかります。

したがって、something.somethingを見ると、これは通常、これがオブジェクトであり、そのプロパティまたはメソッドの1つを利用していることを意味します。

現在、私たちが作成するすべての新しい人間は、15cmの高さから始まります。

今では正しくないかもしれませんか？

そこで、別の人間を作成し、これがジェームズだとしましょう。

また、彼は人間のクラスからも作成されています。これがこの部分の意味です。私たちは人間の工場に行き、新しい人間を作ります。

そして、私が彼の身長を印刷する場合、あなたはそれがまったく同じであることがわかります。

彼は身長も15センチです。しかし、それはおそらく本当ではないでしょうか？

新しい人間が作成されたときに、すべての人が同じデフォルトの高さを持っているのはなぜですか？

これらのプロパティのいくつかについて、各オブジェクトを異なる値で初期化する場合はどうでしょうか？

どうすればいいですか？

前回のレッスンでは、クラスのコンストラクターと呼ばれるものを使用できることを既に見ました。

そして、コンストラクターを作成するために、クラスの名前を書くので、それは人間であり、そして空の括弧のセットを作成します。

関数を作成するとき、これらの括弧に入力を提供するように、クラスのコンストラクタを作成するときに、クラスに入力を提供できます。

したがって、高さプロパティがdoubleであるため、開始高さプロパティはdoubleになる可能性があります。

startingHeightという名前を付けます。

そして今、中括弧内で、これらのstartingHeightを高さに割り当てることができ、高さは値なしで開始できます。 nullから始めます。

しかし、新しい人間を作成するときは、開始時の高さを指定する必要があり、その後、人間は

heightプロパティは、指定した開始時の高さに設定されます。

そのため、ここには開始時の高さが必要であり、それを指定していないため、ここでエラーが発生します。

それで、それを追加しましょう。したがって、ジェニーは15センチメートルから始まる人間であるが、ジェームズは人間であり、彼は少し背が高いかもしれません。たぶん彼は20センチです。

実行をクリックすると、ジェニーの身長が15cmから始まり、ジェームズの身長が20cmから始まることがわかります。

そして、なぜ人間にはデフォルトの身長が必要なのでしょうか？

それらはすべて同じ高さで作成されるわけではありません。また、入力またはパラメーターを名前付きパラメーターにすることができるように、関数ではクラスコンストラクターでも同じことができます。したがって、これらの入力に中括弧を追加して、新しいヒューマンオブジェクトを作成するときに使用する場合、その名前を指定する必要があると言うことができます。

したがって、startingHeight、次にコロン。

このように、目で見ると少し楽になり、あなたは私が新しい人間を作成し、その人間に出発点の高さを提供していることがわかります。前述したように、クラスにはプロパティを設定でき、メソッドも設定できます。メソッドは、クラス内で定義された単なる関数です。

メソッドを作成する場合、たとえば、少し人間に何かを言うように話してみましょう。何を言うかを伝えることができます。

したがって、whatToSayという文字列入力を作成し、次に、クラスに関連付けられた関数について話しているため、talk関数またはこの場合はtalkメソッド内で、文字列を出力するだけであると言えます。と言った。

ですから、たとえば、ジェームズを退治して、新しい人間に伝えることができるということです。ジェニーと言ってみよう

.talkとあなたが言うべきことは、「なぜこの空は青いのですか？」

コードを実行すると、ジェニーの身長が表示され、ジェニーに話しかけることができます。そして、このメソッドtalkを呼び出すことで、このオブジェクトが実際にそのアクションを実行し、コンソールに伝えたいことを出力することがわかります。

これは現在、ヒューマンクラスに関連付けられているメソッドです。

そして、ドット表記を使用してプロパティの高さにアクセスするのと同じように、ドット表記を使用してtalkメソッドをトリガーしています。

したがって、オブジェクトを操作しているときは、あちこちにドットが表示され、「このオブジェクトはこれを行う、またはこのオブジェクトはプロパティを取得します」と言っているだけです。

これで、非常に単純なクラスができました。そして、このクラスは、人間のオブジェクトがどのように振る舞うべきであり、どのような特性を持つべきかを示す青写真です。

そのため、人間のオブジェクトが作成されると、そのすべてが実行できるようになるはずです。もちろん、必要に応じてこれを複雑にすることができます。しかし、本質的には、このオブジェクトがどのように動作するか、またはどのように表示されるかについてのプランを作成しているだけです。そして、新しいオブジェクトとして作成して使用し、必要な数だけこれらを作成し、相互に作用させたり、特定のタスクを実行させたりすることができます。

これがクラスとオブジェクトの作成方法です。

しかし、なぜこれらのクラスとオブジェクトが必要なのでしょうか？それについて調べるには、

次のレッスンでお会いしましょう。