

オブジェクト指向とは

オブジェクト指向には「オブジェクト」「指向」「プログラミング」の3つの特徴があります。

●オブジェクトとは？

人がプログラミングしやすいように、作ったモノ。

クラスやインスタンスなどがオブジェクトに当たります。

●オブジェクト指向とは？

「こういう "モノ" を作りましょう」「そして、その "モノ" を使いましょう」というのがオブジェクト指向という考え方。

●プログラミングとは？

オブジェクトを作って、実際にコードを書くこと。

メリットとしては変更に対応しやすいようにプログラミングできることです。

例えば、サッカーゲームの開発者だとして、まず選手の設定や行動パターンを定義しなければいけません。

そこで下記のように1つずつプログラミングしていったとします。

- DF 選手のシュート力は 50 である。
 - DF 選手は△ボタンを押すとシュートする。
 - DF 選手は□ボタンを押すとパスをする。
-
- FW 選手のシュート力は 100 である。
 - FW 選手は△ボタンを押すとシュートする。
 - FW 選手は□ボタンを押すとパスをする。

しかし開発中のミーティングで下記のように仕様を変更する事になりました。

- 「△ボタンを押すとパスをする」
- 「□ボタンを押すとシュートする」

そのため、これまで書いてきたコードを下記のように1つずつ修正していくことにしました。

- DF 選手のシュート力は 50 である。
 - DF 選手は□ボタンを押すとシュートする。
 - DF 選手は△ボタンを押すとパスをする。
-
- FW 選手のシュート力は 100 である。
 - FW 選手は□ボタンを押すとシュートする。
 - FW 選手は△ボタンを押すとパスをする。

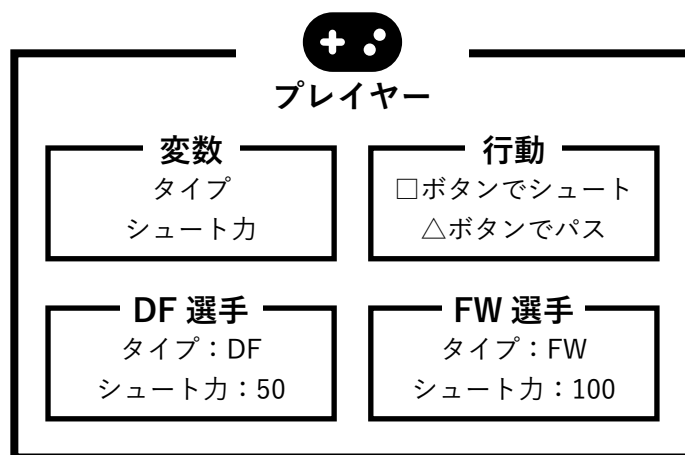
この場合は2人だったので変更にはかかりませんでした。が、選手全員となった場合はどうでしょう。

仕様変更が起きた場合、変更するのに多大な時間や労力がかかり非効率です。
またミスも起こりやすくなり、バグの原因にもなります。

ここで登場するのが「オブジェクト指向」という考え方です。

選手もプレイヤーなので、共通する設定は「プレイヤー」というオブジェクトにまとめておきます。

また選手のタイプによって異なる部分は、「DF 選手」や「FW 選手」などオブジェクトに分けてプログラムを書いていきます。



【プレイヤー】

□ボタンを押すとシュートする

△ボタンを押すとパスをする

シュート力という設定を持つ

DF 選手などのポジションを持つ

【DF 選手】

シュート力は「50」

ポジションは「DF 選手」

【FW 選手】

シュート力は「100」

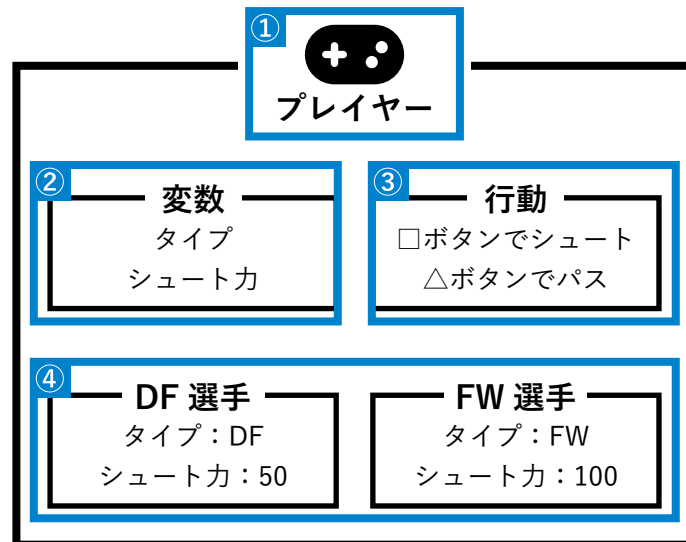
ポジションは「FW 選手」

このようにしておけば「□ボタンを押すとパスをするよう変更したい」と要望があっても、プログラムをオブジェクトで書いておけば「プレイヤー」のコードを書き換えるだけで済みます。

変更に対応しやすいことこそがオブジェクト指向の最大のメリットです。

オブジェクト指向の用途を解説

次にオブジェクト指向を理解する際に重要な用語について「DF 選手」と「FW 選手」の例を用いて解説していきます。



●用語①：クラス

同じ特徴や性質を持つモノをまとめたオブジェクト。

先ほどの例で言うと「プレイヤー」がクラスに当たります。

クラスは、インスタンスを作る為の設計図です。

●用語②：プロパティ

オブジェクトが持っている特徴を言います。

「シュート力」や「ポジション」などがプロパティに当たります。

●用語③：メソッド

オブジェクトが起こすアクションのことを言います。

「□を押すとシュートする」「△を押すとパスをする」がメソッドに当たります。

●用語④：インスタンス

クラスによって、作られたオブジェクトです。

今回の場合、「プレイヤー」クラスによって作られた「DF 選手」と「FW 選手」がインスタンスに当たります。