

1 オブジェクト指向

オブジェクト指向とはプログラムをモノとして扱いモノを組み立てるようにして作るプログラミングの考え方です。

オブジェクト指向を用いることでソースコードを見やすくしたり大人数で開発するときに「オブジェクト指向」という共通の考え方で開発をするので統一性もあります。

オブジェクト指向を説明する上で大切な要素が3つあります。それは「カプセル化」、「継承」、「ポリモーフィズム」です。

ひとつずつ簡単に説明します。まず「カプセル化」です。

カプセル化というのはその名の通りカプセルのようなものに入れて密閉して外から干渉できないようにしたり隠したりする仕組みのことを言います。

これを行うことによって他の人に余計な処理を見せて惑わせてしまうことを防いだり、意図していない使われ方をしないようにできます。

二つ目は「継承」です。

継承とはオブジェクトを引き継いで別の場所で使う仕組みのことです。

例えば車が走るプログラムを作るときに車の色や車種やホイールの種類の見た目を指定するクラスと走る動作をするクラスがあったとします。車1は赤色で時速120kmではしり車2は青色で時速90kmで走る場合車の見た目クラスを一つ作ることによって車1も車2もそれをつかい色だけを変え、その他の条件は一緒にすることができます。走る動作も同じように時速だけ変えれば他の条件は同じにできます。

継承によってプログラムを省略することができます。

三つ目は「ポリモーフィズム」です。

ポリモーフィズムは日本語で多様性とか多態性と言われます。

同じメソッドを複数のクラスに持たせて呼び出すときに呼び出したものによって動作を切り替えられるという仕組みのことです。

例えば剣士のクラスのアタックメソッドでは剣で攻撃する動作、魔法使いクラスのアタックメソッドでは呪文で攻撃する動作、弓師のクラスのアタックメソッドではゆみでこうげきする動作があるとしてそれぞれのクラスでアタックメソッドを呼ぶと動作が切り替わりますよね。

これら3つの要素を使いプログラムを考えることをオブジェクト指向と言います。

2 Git

Gitとはプログラムをネット上やローカルで共有することができるツールです。

Gitを使うにはまずリポジトリを作ります。リポジトリはソースコードが上げられて管理する場所です。リポジトリにはローカルリポジトリとリモートリポジトリの2種類あり、ローカルリポジトリでは自身のPC上でのみ管理することができるものですがソースファイルをコミットするとローカルリポジトリにコミットされたファイルが上がります。コミットするファイルを選ぶことをステージングと言います。リモートリポジトリはネット上で管理することができるものでコミットされたものをプッシュすると上がります。

リポジトリにはブランチといったものがあります。複数人で開発している時それぞれ別の部分をプログラムすると思うのですが担当している部分以外のコードが変わってしまうのは邪魔になってしまいますよね。そんなときにブランチというものが便利でブランチは枝のように枝分かれしていきます。始めにリポジトリを作った時はmasterブランチ一本だけがあります。そこから機能別にブランチを作ります。ブランチでは他のブランチでプッシュされた情報は反映されず同じブ

ランチにいるもの同士でしかプッシュされた内容が反映されません。このようにして複数人での開発をより便利に行うことができます。