**Strategyパターン**

日付: 2015年01月23日

　　　　　　　　　　　　　　　　場所 Swift　19:30 ~ 20:30

参加者: 関根さん、新井さん、香取さん、辻くん、ちくりん、ハー、アインくん、クオン

⚫定義

Strategy とは英語で「戦略」を意味する言葉です。Strategy パターンを利用することで、戦略の切り替えや追加が簡単に行えるようになります。

プログラミングするとき、関数の中に複雑なアルゴリズムを実装して、if-else分岐を使ってしまうことが多いです。Strategy パターンでは、戦略の部分を意識して別クラスとして作成するようにしています。戦略と部分を別クラスとして作成しておいて、戦略を変更したい場合には、利用する戦略クラスを変更するという方法です。

すると、アルゴリズムより柔軟に変更、追加しやすいです。

⚫️分かりやすくて、結果が良く見れるし、どんなプログラミング言語にもよく使われています。

⚫ReFUEL4プロジェクトで自分がStrategyパターンを使用したことがあります。（ここのコード、説明等を勉強会の前に話しました）

Geckoboardサービスで使われています。

Geckoboardサービスで多数なグラフタイプを提供していて、グラフごとに異なるデータタイプを準備しないといけません。

将来的に、グラフのタイプを変更、追加、データタイプの変更などの可能性が高いので、スムーズに行くため、Strategyパターンを使っています。

⚫️定義、説明が難しくなくて、普段なプログラミングでもよく使えるパターンです。

個人的に、これが好きです。

⚫参考リンク

<https://sourcemaking.com/design_patterns/strategy>

<http://liginc.co.jp/web/programming/php/136131>

<http://www.techscore.com/tech/DesignPattern/Strategy.html/>