

# フローチャート

プログラマはプログラム(ソースコード)を作成する前に、要求仕様(課題)を理解し、その要求をどのように実現するかを考えます。所謂、設計作業です。頭の中だけですぐに考えがまとまる場合もあれば、まとまらない場合もあります。考えをまとめる時には図やメモを使うと、考えを整理しやすくなります。その図やメモの書き方も何らかのルールに従うようにしておく、他のメンバーとレビューを行うときや、設計情報として残しておくときに都合がよく、情報としての価値も上がります。書き方のルールはチーム内で決めるだけでも良いのですが、できれば世界中のエンジニアに広く知られているルールの方が効果的でしょう。広く知られた書き方のルール(記法)の一つがフローチャートです。数ある記法の中でもっとも古くから使用されている記法です。実際の開発作業で、作成するすべてのプログラムをフローチャートで書き示すことはほぼ無いと思いますが、フローチャートを使うことで問題解決につながる場面は多くあると思います。

フローチャートには多くのダイアグラムがありますが、それらを丸覚えしようとしたり、ダイアグラムを数多く使おうとするのは間違いです。書き方は必要なときに調べることができれば全く問題ありません。フローチャートの使いどころを会得することが重要です。

## 参考記事

「若手プログラマー保存版！フローチャート徹底解説と作成カンニングペーパー」

URL [https://www.ntt.com/business/sdpf/knowledge/archive\\_47.html](https://www.ntt.com/business/sdpf/knowledge/archive_47.html)

## 練習課題

1から5までの番号が付いた5枚のカードを番号の大きい順に並べるための処理をフルーチャートで書きます。

