

密結合と疎結合

tight or loose coupling

ソフトウェア研究会in秋葉原

池田公平 2013/8/20

密結合とは？

- ◆ 密結合とは、個々のコンポーネントやクラスが密接に関わり合っている状態を示す。
- ◆ 密結合は、機能の実装や効率化がやり易いという長所がある。
- ◆ いっぽうで、機能の変更やコンポーネントの追加が困難なこと、結合相手の不具合による影響を受けやすい、単体のテストが難しいといった短所がある。

疎結合とは？

- ◆ 疎結合とは、コンポーネントやクラス間の依存性が少なく、個々が独立した状態で動作しやすい状態をいう。
- ◆ 疎結合は、機能の追加や修正、再利用、リファクタリング、テストがやり易いという長所がある。
- ◆ いっぽうで、**API**設計をしっかりと行わないと実現が困難なこと、実装を進めていく上で、常に疎結合を意識しないと密結合に陥ってしまうという短所がある。

人間の世界に例えると？

仲の良いカップルは密結合です



➤何をやるにも一緒。常に二人三脚で行動します。

➤お互いに助け合うので、困難な局面にも対処できます。

➤二人三脚といっても、一人よりも速く走れます。

➤相手のことを熟知しているので、相手の力をいつでも利用できます。

仲が良いのは良いことだけではない



- ◆ どちらかが病気になると、看病のため何もできなくなります。
- ◆ 浮気はできません。より良いパートナーが見つかったても、乗り換えは困難です。
- ◆ 3人以上で協力して作業する場合、人間関係に問題が生じる可能性があります。

両手に花状態は疎結合です



- ◆ 個々との係わり合いが浅いので、一度に何人でも付き合えます
- ◆ より良い相手が見つかった場合、いつでもパートナーに追加できます。
- ◆ 相手が多いので、多人数で行う仕事も難なくこなせます。
- ◆ 誰かが病気になっても、ほとんど影響を受けずに済みます。

疎結合は付き合い方に注意が必要



- ◆ すべての相手とバランスよく付き合わないと、競合が起こるかもしれません。
- ◆ 誰かと仲良くなりすぎると、他の人が離れてしまい、密結合になってしまいます。
- ◆ 一度に多くの人と付き合い方には、それなりのスキルが必要です。

もっとも密結合が強いのは親子です



- ◆ どんなに仲の良いカップルも、親子の縁の強さにはかないません。
- ◆ 子は親のすべてを見て育ちます。
- ◆ 親の良い所も悪い所も継承します。

親子の関係が深すぎると



- ◆ 親が子供の世話を焼きすぎると、子供は一人ではなにもできなくなります。
- ◆ マザコンになってしまった子供は、親への依存度が強すぎて、自分の能力を発揮できません。
- ◆ 親は、必要以上に子供の世話を焼かず、進むべき道を示すだけにしておいたほうが、良い親子関係が築けます。

オブジェクト指向の世界も同じです

- ◆ 最も結合が強いのは、「is-a」である継承関係にあるクラスです。
- ◆ 親の機能が多ければ多いほど、結合が強くなり、子の負担が増えます。
- ◆ 多重継承や継承関係を乱用すれば、簡単に密結合の世界に陥ります。
- ◆ 何でも助けてくれる親よりも、方向性を示して子供に作業を任せる親のほうが、優れた親です。

コンピューターの世界では疎結合を築こう

- ◆ 人間社会では、密結合も良いですが、コンピューターの世界ではハーレム状態の疎結合のほうが優れています。
- ◆ 依存度を下げることで、自由度が広がり、多くの人と関わり、結果的に大きな力を得ることができます。
- ◆ 最初は疎結合でも、密結合に陥ることがあるので、注意が必要です。

疎結合を実現するには設計が大切

- ◆ 疎結合を実現するには、機能ごとにコンポーネントを分離した設計が不可欠です。
- ◆ 各コンポーネントの依存関係を図にすることで、結合度を可視化します。
- ◆ 強い結合や循環的な依存がある場合は、設計の見直しが必要です。
- ◆ 機能変更や機能追加、実装段階での作りこみで、疎結合が崩れていないか定期的なチェックが必要です。

密結合の例

