

# オブジェクトシステム'02 # 1

久野 靖\*

2003.1.22

## 1 はじめに

- 科目「オブジェクトシステム」の趣旨…
  - 世の中ではオブジェクト指向プログラミングが一般化/普及
  - オブジェクト指向プログラミングとは (Java でやったはずね?)
  - それを使いこなすにはどういうことが重要? どうすればいい?
- さまざまな切り口が考えられるが…
  - とりあえず「Java Design」を読んでみようかと
  - 時間が余るようならまたそれなりに題材を考えます

## 2 Design(設計) とは?

- ソフトウェア (プログラム) の「設計」→作る前に構造等を決める
  - 「設計」しないでソフトを作ることは?
  - 「設計」って具体的にどんなことをやるの?
  - 使用する言語によって「設計」の内容は違うの?
- オブジェクト指向言語→どういうオブジェクトをどう組み合わせるかが「設計」
  - 多くの言語ではオブジェクトはクラスのインスタンス
  - →クラスの設計、オブジェクトの組み合わせ方の設計
- そもそも、どうやってオブジェクトを決めるの?
- そもそも、どうやって組み合わせ方を決めるの?

## 3 設計技法

- ソフトウェア設計を行う手法。いろいろある。
- オブジェクト指向以前…
  - 構造化設計
  - ジャクソンシステム設計方 (JSD)
- オブジェクト指向言語の登場→従来の設計技法では合わないことが多い
  - クラス、継承などを適切に設計するのが難しいが…
  - これらは旧来の設計技法には (当然) 現われない
- オブジェクト指向に向けた設計法が複数提案されてきた
  - Booch 法
  - OMT 法 (Ranbaugh)
  - Yordon
- バラバラでは困るの統一の必要性
  - Booch が Ranbaugh のいる Rational 社に移籍したのがきっかけで…
  - 統一プロセス (Rational Unified Process、RUP)
- この授業はソフトウェア工学ではないので RUP とかは扱わない
- テキスト「Java Design」→もっとフランクにこの手の手法を見せてくれる良書だと思うので取り上げた

## 4 UML(Unified Modeling Language)

- もともとは図の描き方も手法ごとにバラバラ
  - →それでは A 氏が描いた図が B 氏には見にくい→統一の必要性

\*筑波大学大学院経営システム科学専攻

- →共通の図法を制定して UML という名前をつけた
- 従って UML そのものは特定の手法には依存していない
  - 今回の目的にとっても（「Java Design」にとっても）好都合
- UML そのものはさまざまな情報を表現するためにどんどん拡張され詳細化
  - これについては扱わない。あくまで必要な場合に UML の記法を使うだけ

## 5 Java Design

- では 1 章から始めてみましょうか…
  - どういう感じになるかはやってみないと分からない :-)