

2013 ソフトウェア工学

2013年4月10日

海谷 治彦

目次

- 講師陣
- 講義の目標
- 単位認定について
- 教科書
- 授業の進め方
- 第一章 なぜプログラミングは難しいか
- 第二章へのイントロ
- Java, UML, Eclipse
- 受講上の注意
- 次回 4/13水の演習について

講師陣

- 海谷先生
私
- 坂下センパイ
Teaching Assistant (TA)
- 湯浅センパイ
Teaching Assistant (TA)
- 小形先生
 - たまに代打でいてくれるかも.

講義の目標

- ソフトウェアを設計してから、プログラムを開発するような人に受講生がなること。
 - いきなり、エディタ(もしくはIDE)でコードを書くのは今後はNG.
- 特にオブジェクト指向設計ができること.
- Javaでオブジェクト指向プログラムが書けること.
- UMLでオブジェクト指向設計ができること.

単位認定について

- 演習(プログラミングやモデリング)の結果で判断します.
 - 大体, 4回前後.
- 中間試験や期末試験は,
 - 様子を見てどうするか決めます.
- 出席は・・・
 - 大学の命令でたまにとります.
 - 参加することに意味は・・・

教科書

- 古い本ですが、とてもしつかりした考え方を平易に伝えています。
- Javaの文法は古いので、授業中に適宜補います。
- 上級者と思っている人にも学ぶべき所のある本です。
- 題名が残念。



授業の進め方

- 基本的に教科書にそって講義やります。
- 教科書の古い部分, 不足している部分を適宜, 補います。
- 演習をやってもらいます。
 - プログラム, 設計, たまには感想や考察。
- 特に発表会とかは考えてませんが, やりたい人がいれば名乗り出てください。
- スケジュール: 以下から確認してください。

<http://kaiya.cs.shinshu-u.ac.jp/se/>

ソフトウェア工学

ってか、情報工学って何？

言葉の定義 by 広辞苑

- 工学:
基礎科学を工業生産に応用して生産力を向上させるための応用的科学技術の総称。
- 工業:
原料や粗製品を加工して有用なものとする産業。
- 生産:
自然物に人力を加えて、人にとって有用な財を作り出し、もしくは獲得すること。
- 産業:
生産を営む仕事、すなわち自然物に人力を加えて、その使用価値を創造し、また、これを増大するため、その形態を変更し、もしくはこれを移転する経済的行為。

言葉の定義 by 広辞苑

- 工学:
基礎科学を工業生産に適用し、向上させるための応用科学。
- 工業:
原料やエネルギーを加工し、製品を生産する行為。
- 生産:
自然物から人工物への変換、もしくは既存の人工物を加工して新たな人工物を生み出す行為。
- 産業:
生産を営む行為を指す。生産を加えて、その使用価値を増大させるため、その形態を変更し、もしくはこれを消費する経済的行為。

工業・工学の根幹は主観的

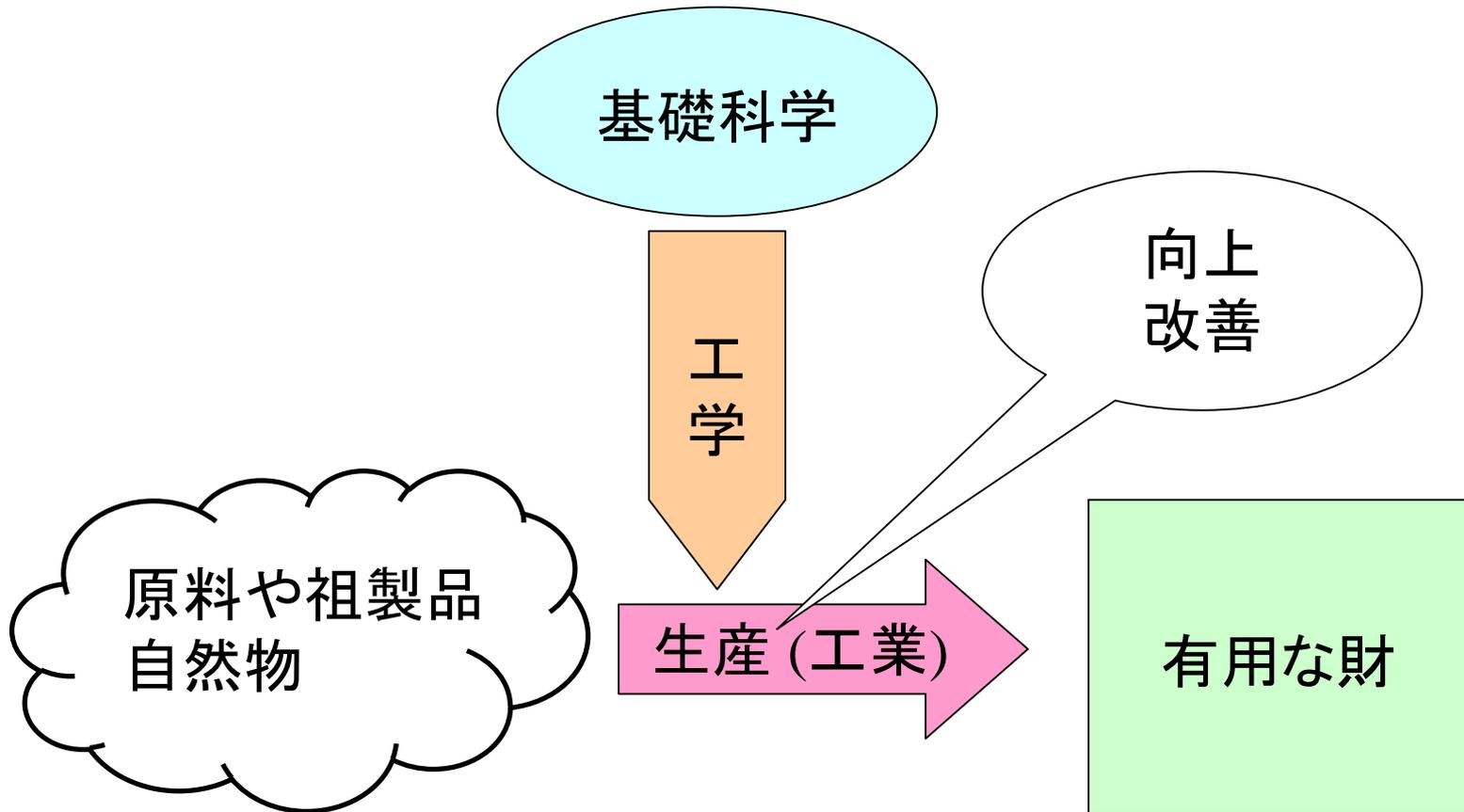


ヒトによる
有用性, 価値に
依存している。

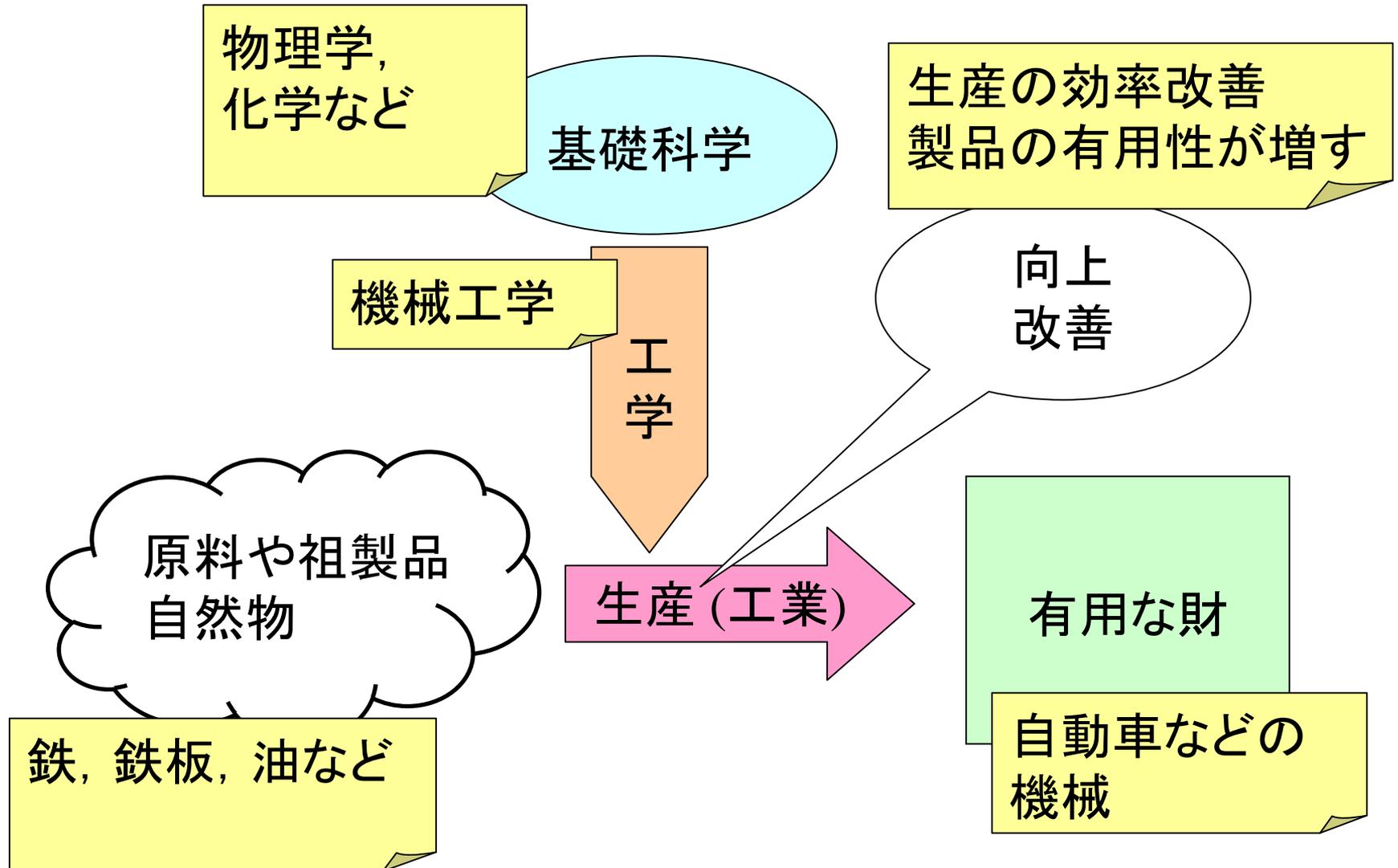
科学 by 広辞苑

- 世界と現象の一部を対象領域とする、経験的に論証できる系統的な合理的認識。
- 研究の対象あるいは方法によって種々に分類される
 - (自然科学と社会科学、自然科学と精神科学、自然科学と文化科学など)。
- 通常は哲学とは区別されるが、哲学も科学と同様な确实性をもつべきだという考えから、科学的哲学とか、哲学的科学とかいう用法もある。

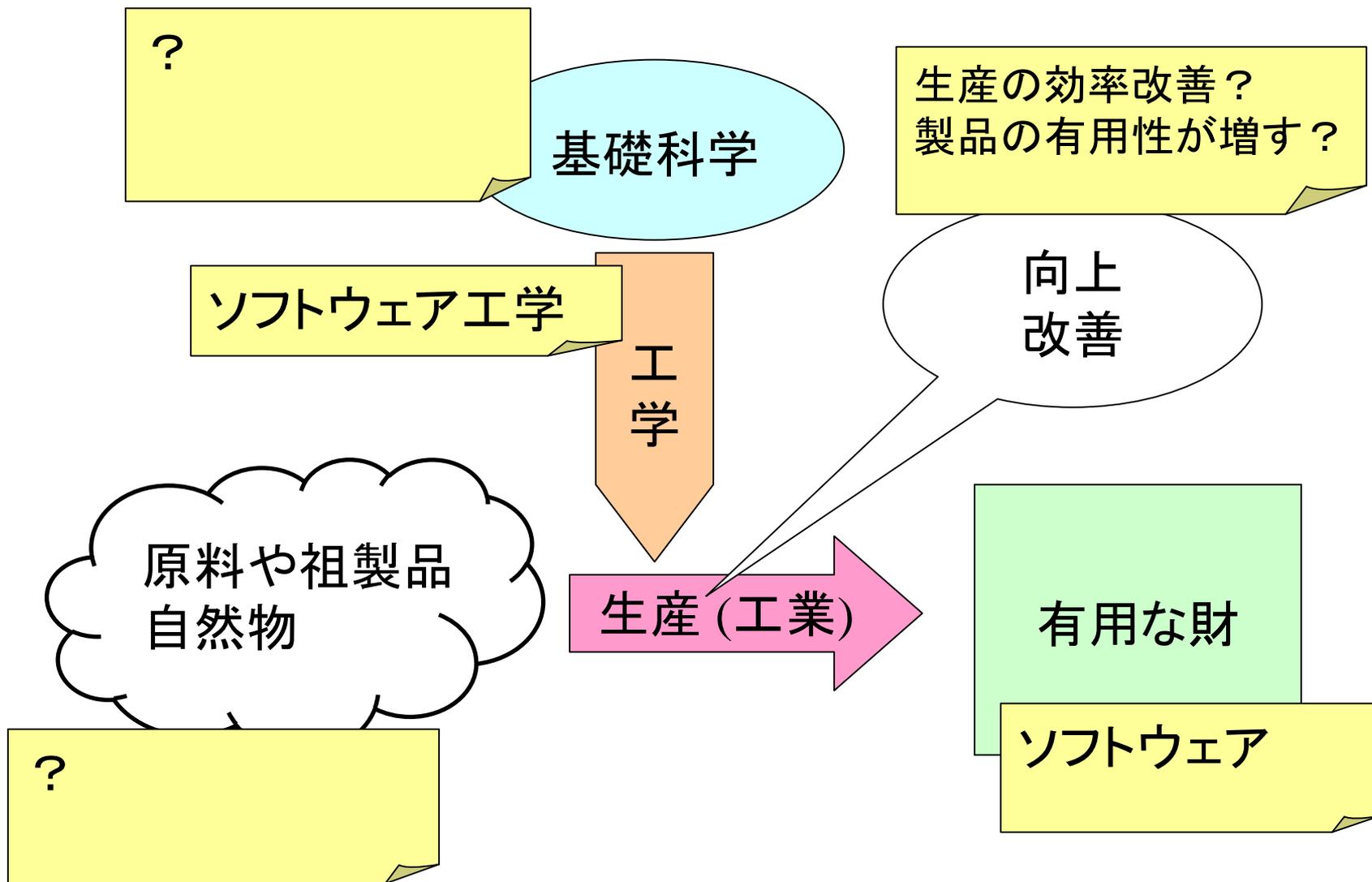
工学



工学と機械工学



工学とソフトウェア工学



教科書 1章について

第一章の結論

- なぜプログラミングは難しいか？

以下のどこかでつまづいている！

1. コンピュータに行わせたいことを**理解**
2. 理解したことを説明できるレベルまで**整理**
3. コンピュータにわかる言葉に**翻訳**

プログラミングとは何か？

- コンピュータにやらせたいことの手順を、コンピュータのわかる言葉で書く。
- 教科書 15～16ページの掛け算の例

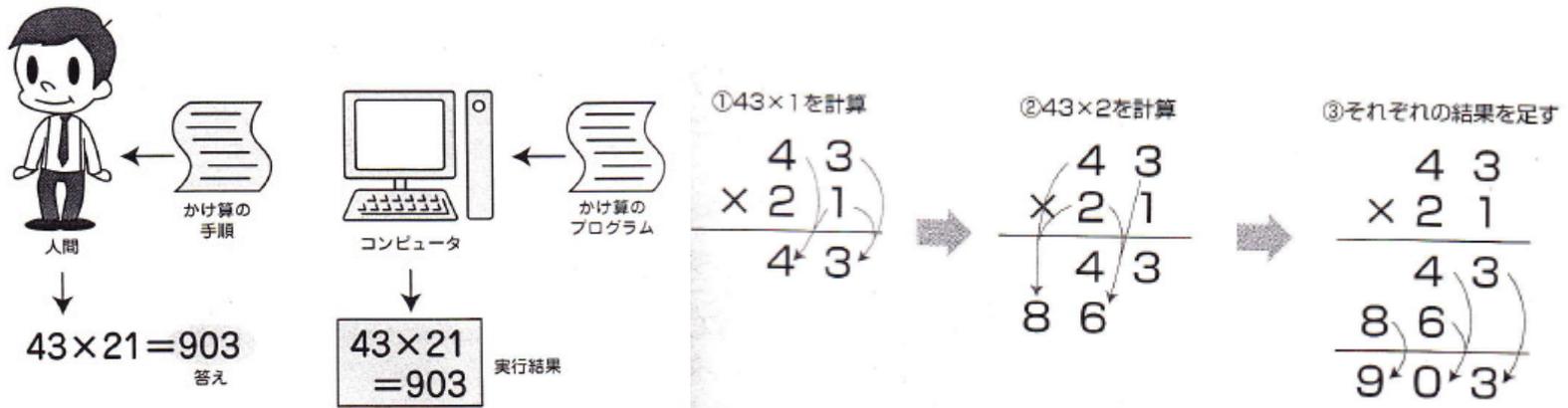


図 1-2 掛け算とプログラムの対比

理解の失敗例

- もし、以下にあるような掛け算の手順を追
い、意味が分からなければ、掛け算のプロ
グラムはできない！

① 43×1 を計算

$$\begin{array}{r} 43 \\ \times 1 \\ \hline 43 \end{array}$$

② 43×2 を計算

$$\begin{array}{r} 43 \\ \times 2 \\ \hline 86 \end{array}$$

③それぞれの結果を足す

$$\begin{array}{r} 43 \\ \times 21 \\ \hline 86 \\ 903 \\ \hline \end{array}$$

- もし、銀行業務が理解できなければ、その
業務支援ソフトウェアは作れない。

何故，考案ではなく理解か？

- コンピュータにやらせたいことの多くは，現実世界の業務や手順の一部である。
 - 放射性物質の飛散予測の計算の一部（全部）
 - 銀行業務の一部
- そのような業務や手順は，その道の専門家が考案する。
 - 原子力専門家，物理学専門家，気象学専門家
 - 銀行員
- 我々，コンピュータ技術者は，これらを理解し，計算機で(どれだけ)肩代わり可能か判断する。
 - 判断のためには理解が必須！

整理の失敗例

- 個々の掛け算の計算手順はわかるが、それを一般化(整理)して、他人に説明できない。
 - 例えば、算数や数学にあるように N や x みたいなパラメータを使って、計算手順を一般化できないとか。
- 銀行業務も個々の決済等の業務を一般化した手順として書けなければ、ソフトウェアを作れない。

整理のポイント

現実業務と計算機が可能なことのバランスが重要

- 現状の計算機(プログラム言語)で実現不可能な整理をしてもシステムはできない。
 - 微分方程式で飛散予測を整理できても、それをコンピュータで直接実行はできない。
- 業務と大きく剥離した形で整理しても、そもそも整理されているか確認しようが無くなる。
 - Cプログラムを直接見せられても分からない物理学者も多いだろう。
 - 結果として、計算機にやらせたいことが整理されているか、確認しようが無くなる。

説明相手は誰？

基本的に相手は二種類を想定する。

- 計算機

- 計算機にやらせることを想定しているのだから、計算機に説明することを想定しないと。
- 曖昧さや、直観は通じない。

- そもそもの業務専門家

- コンピュータ技術者の理解が合っているかは、往々にして、専門家じゃないとわからない。
- 専門家は人間だから、基本、長大なプログラムやアルゴリズム記述の意味はわからない。

整理に創造は必要か？

- 現実的には、どう整理するかについて、新しい整理法や整理様式を新規に創造する必要は無い。
- 理解したことを、既存の様式を真似て、整理する技術をまずは覚え使えるようにすること。
- 例
 - PCのフォルダや組織階層等、階層的な構造(木構造)を整理するにはよく知られた様式が既に存在する。(コンポジットパターン)
 - この様式を超える整理法を創造するのは多分無理。
- 既存の整理法を数多く知った上で、より良い整理法を創造することを最終的には目指してほしい。

翻訳の失敗例

- ずばり, プログラム言語を知らない.
 - この辺を補う図書は腐るほど出版されている.
 - 所謂, プログラム言語の授業は結構, この辺のみが重視されている.
- 言語を知っていても, 一般化した手順の記述との対応が分からない.
 - 配列, リスト, スタック, 木等は知っていても, それが「整理されたコンピュータにやらせたいこと」の何に対応するか分からない.
- 日本語を知っていても, コミュ力が無い人が居るのと同じ.

本当に難しいのは理解と整理

- 理解

- 株取り引きの業務が理解できてますか？
- ゲーム内での3D表示の人間の描画法を理解できてますか？
- SPEEDI (スピーディ)の放射性物質の飛散予測法を理解できますか？
- 迷惑メールのみを除去する手法を理解できますか？

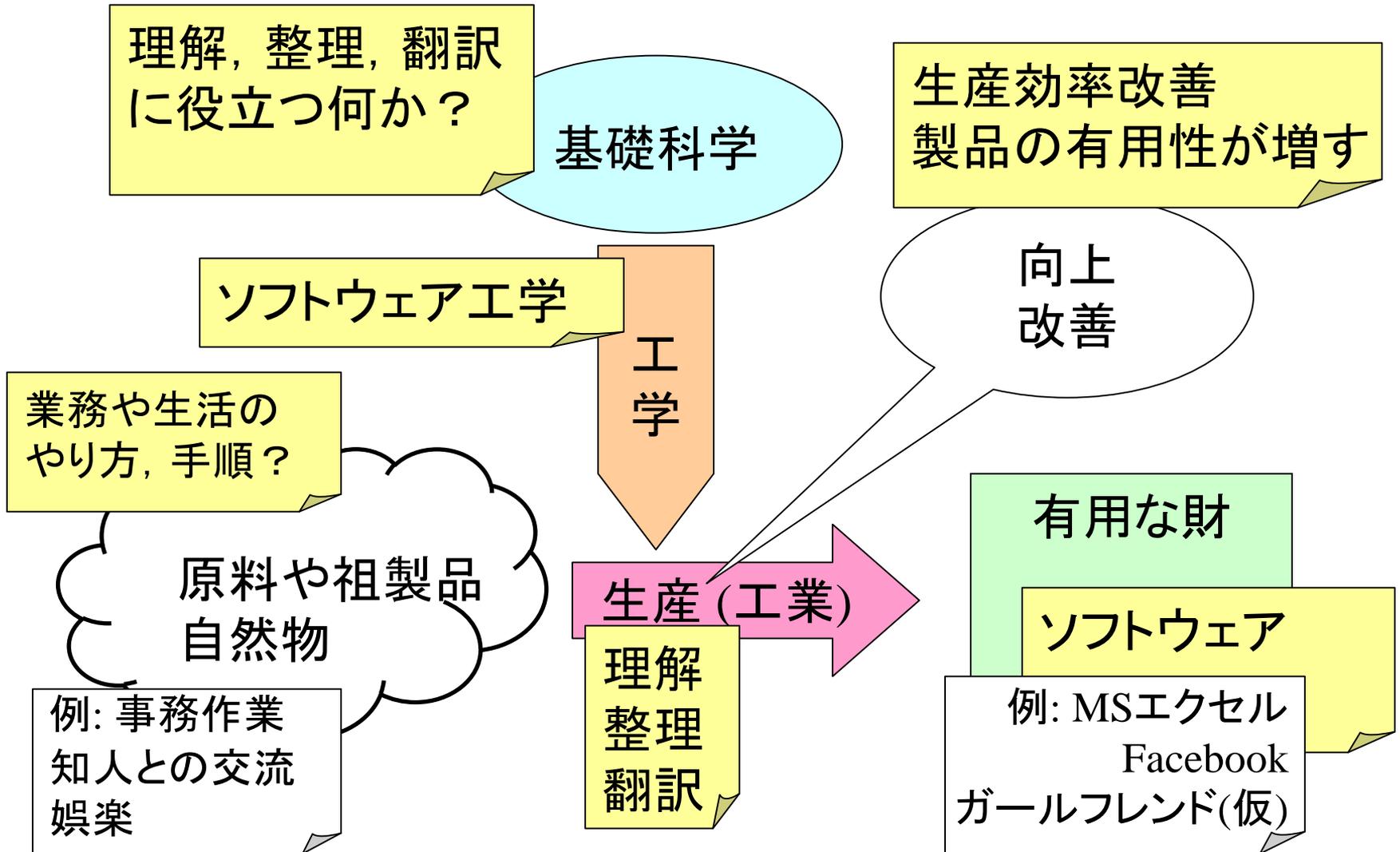
- 整理

- 上記を個々の事例ではなく、一般化して整理できますか？(整理する既存法を知ってますか？)
- 整理したことを人間と計算機の双方に都合よく説明できますか？

計算機屋は下請けか？

- 大筋で YES
 - 対象分野の一部もしくは全部を肩代わりする下請け業務.
- しかし、計算機技術により業務・生活の方が変化することも最近が多い.
 - 技術主導で世の中のあり方・やり方が変わる.
 - 例
 - 検索技術の発達による情報整理法の変化.
 - 図書館等 VS Webやデスクトップ検索
 - 携帯端末の発達による待ち合わせ法の変化.
 - いまどき, きっちり場所と時間を指定しなくても集まれる.
 - SNSの発達により, グループ, 伴侶, 家族等の在り方の変化

工学とソフトウェア工学



二章へのイントロ

オブジェクト指向の利点

- 理解対象である現実世界の事柄(株取引やゲームソフト)を理解や整理するのに、従来のやり方よりはマシである。
 - 従来のやり方: データ構造を作り、それを関数等で処理するC言語的な方法.
- 従来手法よりは、作ったものを改造しやすい.

詳細は次回に.

Java と UML

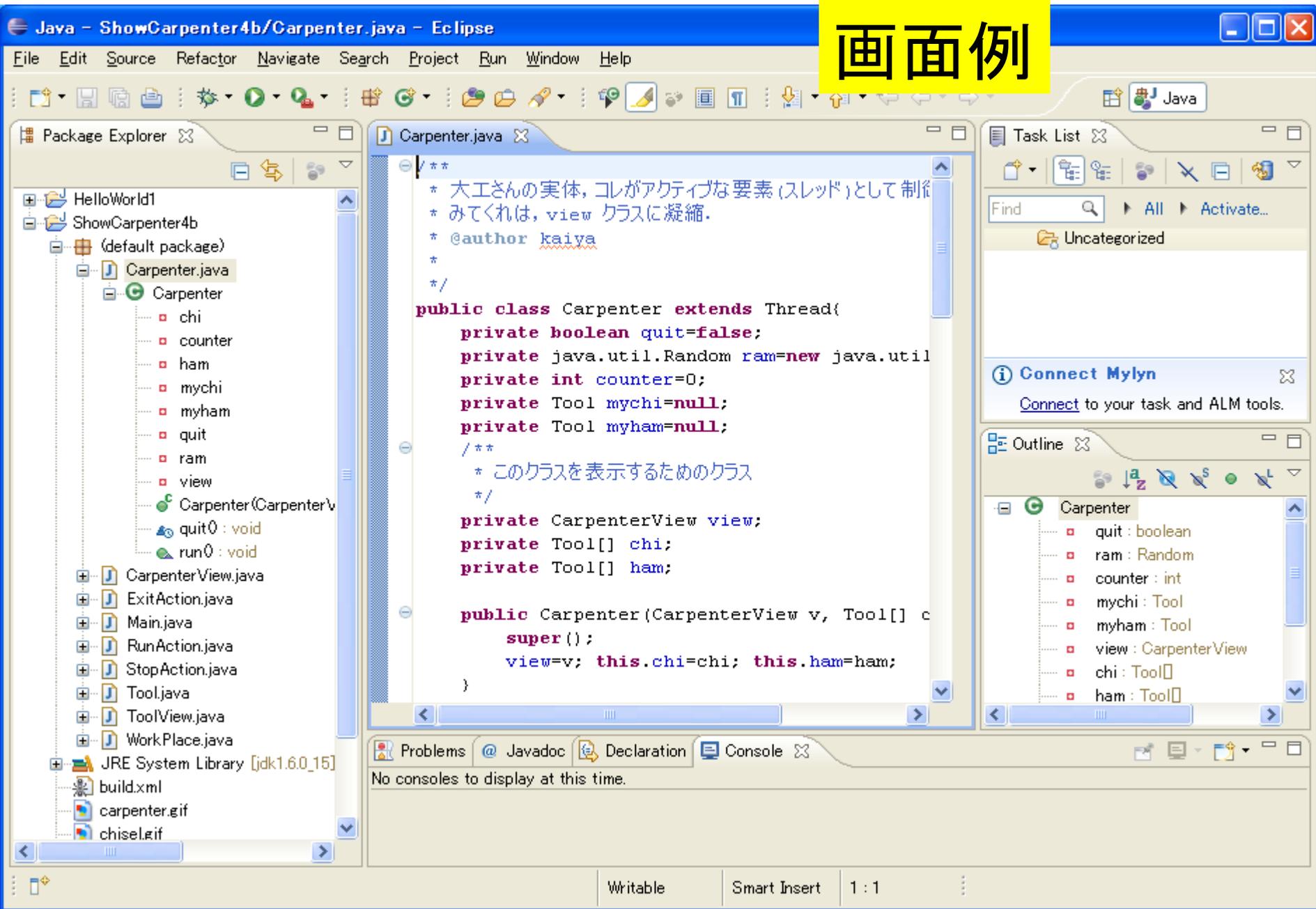
- Java
 - C言語と同様, プログラミング言語です.
 - 代表的なオブジェクト指向言語です.
 - 発案した会社はオラクルに食べられました.
 - 上記の事情で今後が不安 (JDK7でやらかしてる)
- UML
 - 前述のコンピュータにやらせることを整理する道具.
 - というか, 整理した結果を書く道具か.
 - ソフトウェア(プログラム)を設計するための図の書き方.
図式言語.
 - ソフトウェアの設計書としては今日一般的.
 - フローチャートとかの親戚と当面思ってください.

Eclipse えくりぷす

- プログラムを開発する専用のプログラムの一種.
- Integrated Development Environment (IDE) 統合開発環境と総称されるものの一つ.
- Eclipse はJava専用というわけではありません.
- マイクロソフトのVisual Studio もIDEの一種.

- 昨今のプログラム開発では, 2年生以前の演習のように, テキストエディタ, コンパイラを生で使うことは稀で, 通常, IDEを使う.
- どう使うか, どう便利かはおいおい解説していきます.

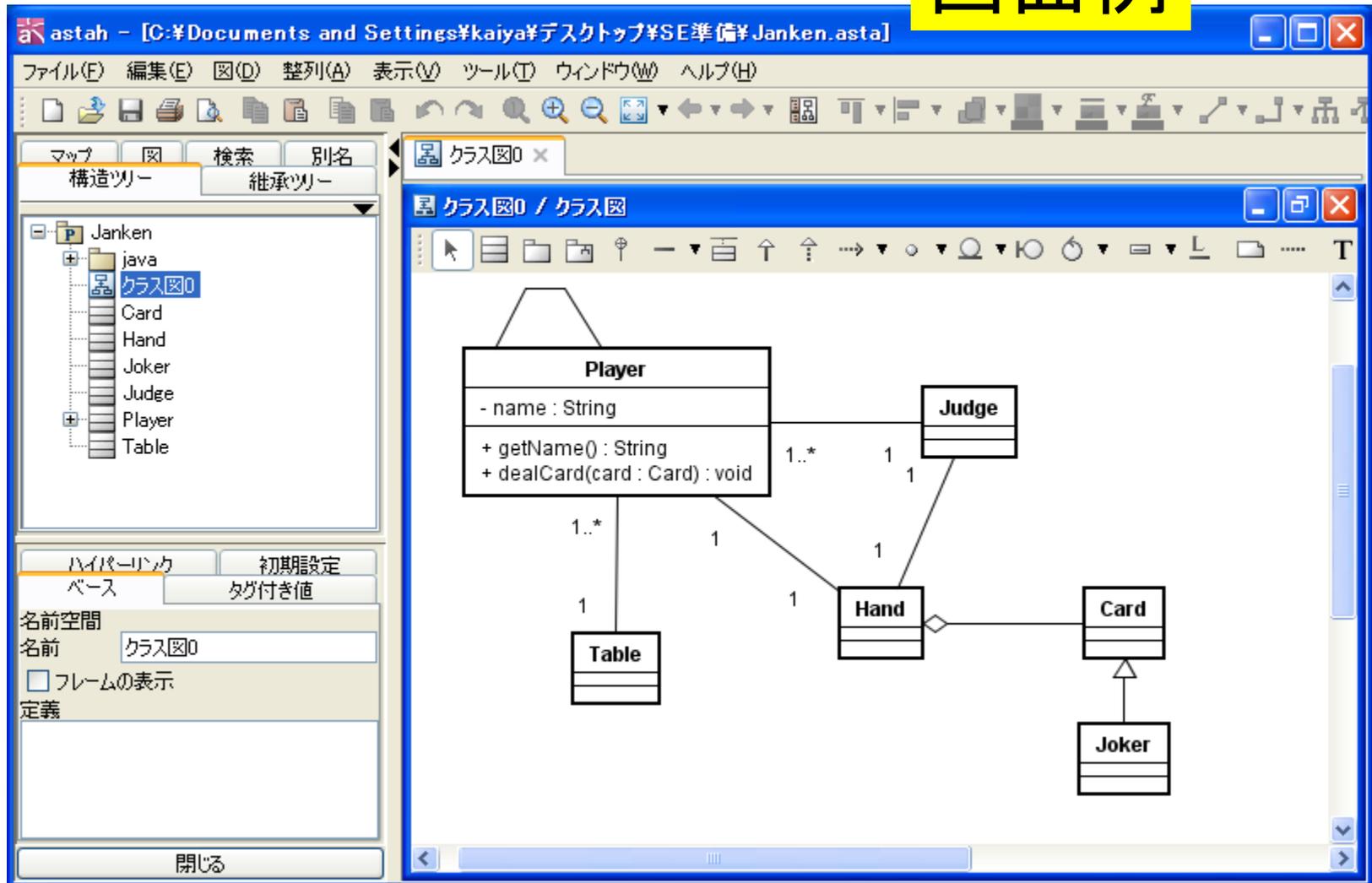
画面例



astah

- UMLを描くためのお絵かきツール
- どのようなソフトにするかの設計をする際に利用する.
- この分野では珍しく made in Japan!
- 実装に近い形でソフトを設計すれば, コードのひな型も生成してくれる.
- 概念的な設計にも利用できる, 例えばビジネスモデリング等.

画面例



受講上の注意

- 受講登録を行い, eALPS からソフトウェア工学のページを参照できるように**必ず**しておいてください.
- 自分のラップトップ(ノートパソコン)は必ずもってきてね.
- 出席はたまにとるかも.

次回の演習

- 開発環境EclipseとUMLモデリングツールastahのインストールと動作テストを行なってもらいます.
- EclipseのサイトやeALPSから必要なインストーラーやライセンスをダウンロードして勝手に進めて結構です.
- Java SDKのインストールも行います.
 - 特に javadoc のため.
- 全体の習熟度を知りたいので全員参加してください.