

Elements of Programming プログラミングの基礎

第 13回 2004年1月12日

Murata Tadahiko

村田 忠彦

<http://www.res.kutc.kansai-u.ac.jp/~murata/>

今回の内容

1. オープンソースのすすめ
2. 変数 (6.1節)
3. データ構造とアルゴリズム (6.3, 6.4節)
4. 関数
5. モジュール化
6. ソフトウェアエンジニアリング (10.1, 10.2節)
7. 構造化手法 (10.3節)
8. オブジェクト指向
9. ソフトウェアの品質 (10.4節)
10. 保守しやすいソフトウェア (10.5節)

Copyright (C) 2001-2004. Murata.

2

構造化プログラミング (p.536)

- 構造化プログラミング
 - ダイクストラ (E.W.Dijkstra) により提唱される.
 - 構造化定理
 - どのような制御を持つプログラムでも, 接続, 選択, 反復の 3つの基本構造の組合せで表すことができる.

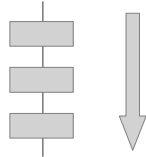
Copyright (C) 2001-2004. Murata.

3

構造化プログラミング

- 接続

- 記述された文の順に実行するというもの . 多くのプログラム言語では , 単に文を順に記述すればその順に実行される .



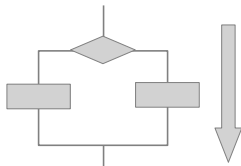
Copyright (C) 2001-2004. Murata.

4

構造化プログラミング

- 選択

- 条件によって実行が分岐するもの . 通常のプログラム言語では , if 文などが用意されている .



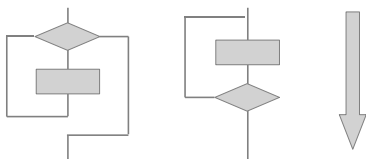
Copyright (C) 2001-2004. Murata.

5

構造化プログラミング

- 反復

- ある条件の下で実行を繰り返すもの . プログラム言語では , ループがこれに相当する . 繰り返しの条件判断を先に行う前判定反復と後で行う後判定反復がある .



Copyright (C) 2001-2004. Murata.

6

手続き型からオブジェクト指向へ

- 従来のプログラミング (手続き型)
 - 処理に重点
 - コンピュータ内部のCPU (Central Processing Unit: 中央演算処理装置)が扱う処理の形式を意識したプログラミング
 - 問題点
 - ハードウェアの仕様や業務の流れが変化したときに処理機能中心のプログラムは変更点が多い。

Copyright (C) 2001-2004. Murata.

7

手続き型からオブジェクト指向へ

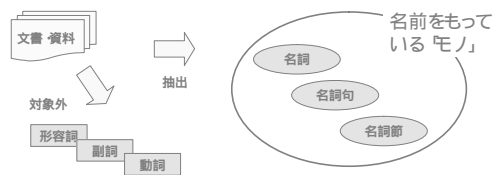
- 手続き型
- オブジェクト指向
 - アプローチ手法の転換
 - オブジェクト指向では、対象となるオブジェクト (データの固まり)を中心としたアプローチをとっていて、柔軟性の高いシステム構築を可能にしている。

Copyright (C) 2001-2004. Murata.

8

オブジェクトとは

- 定義: 感覚によって知覚できるもの、物質、わかりやすい心で知覚できるもの。
- 人間の実生活において存在するペンや手帳、机、いすなどの「モノ」を指す。



Copyright (C) 2001-2004. Murata.

9

プロパティ

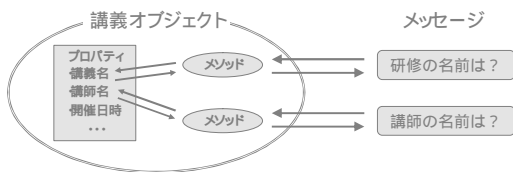
- プロパティ:オブジェクトに含まれる情報
 - 「プログラミングの基礎」オブジェクト
 - 講師 村田,対象:2年次以上,開催時間 水曜 3限
 - 分類 選択科目
- プロパティの隠蔽
 - 外部からプロパティが変更されると,オブジェクト内部での整合性が崩れるため,オブジェクトはプロパティの内容を公開しない.
 - その代わりに,外部からの問合せに対して回答する.

Copyright (C) 2001-2004. Murata.

10

メソッド

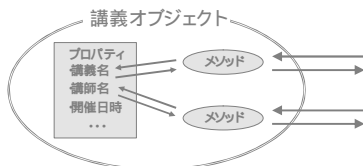
- メソッド:オブジェクトへの問合せに対する回答
 - 講義」オブジェクトのメソッド
 - 講義の名前は?」講師の名前は?」という問合せに対して,自身のプロパティを参照して,回答する.



Copyright (C) 2001-2004. Murata.

11

カプセル化と情報隠蔽



- プロパティとメソッドの外部からの独立化
 - オブジェクトの持つデータや状況の変化は,オブジェクトが自立的に管理できる.このことをカプセル化という.カプセル化により,情報隠蔽が行われる.
 - プログラムの関数化の際と同様の考え方.

Copyright (C) 2001-2004. Murata.

12

クラスとインスタンス

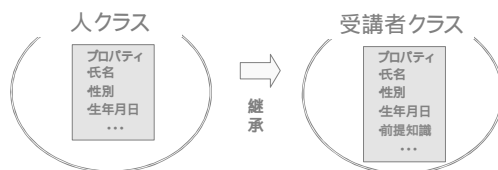
- 「講義群」と「個別の講義」
 - インスタンス (instance) …「個別の講義」
 - 個々の講義のプロパティの種類は同じだが、「講義名」や「講師名」など具体的な内容が異なっている。これらの個々の講義をインスタンスと呼ぶ。
 - クラス (class) …「講義群」
 - 同じプロパティをもつインスタンスの集合
 - オブジェクト(object)
 - クラスとインスタンスを区別しないような場合

Copyright (C) 2001-2004. Murata.

13

継承 (インヘリタンス)

- 継承 :クラスのプロパティを受け継ぐこと
 - 「受講者」クラスは、「人」クラスを継承している。

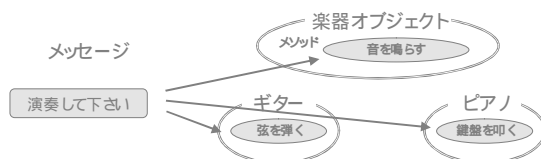


Copyright (C) 2001-2004. Murata.

14

ポリモルフィズム

- ポリモルフィズム :多様性,多態性.
 - 同じ1つのメッセージを送るだけで、メッセージを受け取るオブジェクトがそれぞれ自分のメソッドを使ってその要求を果たすこと。



Copyright (C) 2001-2004. Murata.

15

オブジェクト指向

- オブジェクト指向
 - プロパティとメソッド
 - クラスとインスタンス
 - インヘリタンス :クラスの継承
 - ポリモルフィズム :同一メッセージによる振舞
- 目的
 - 情報隠蔽
 - 継承による再利用
 - ポリモルフィズムによる一貫性

Copyright (C) 2001-2004. Murata.

16

保守 (p.541 ~)

- 保守とは
 - 何らかの理由により,変更 修正すること
1. 修正保守作業 バグ
 2. 変更保守作業 環境変化
 3. 改良保守作業 機能追加 ,操作性向上

Copyright (C) 2001-2004. Murata.

17

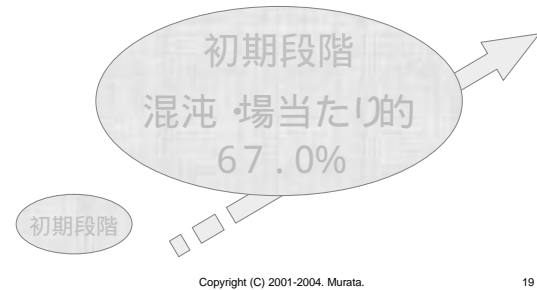
保守を意識した設計

1. 適切なモジュール分割
2. 共通部分のくりだし
3. 適切なモジュールサイズ
4. 規約にあったコーディング
5. わかりやすい変数名
6. 理解を助けるコメント
7. 十分なドキュメンテーション

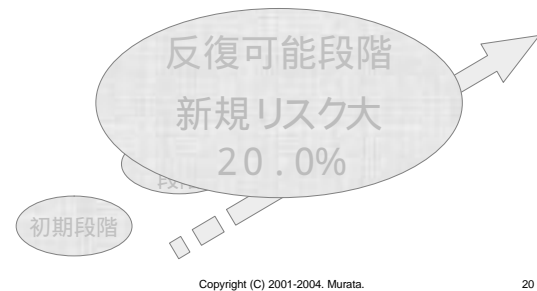
Copyright (C) 2001-2004. Murata.

18

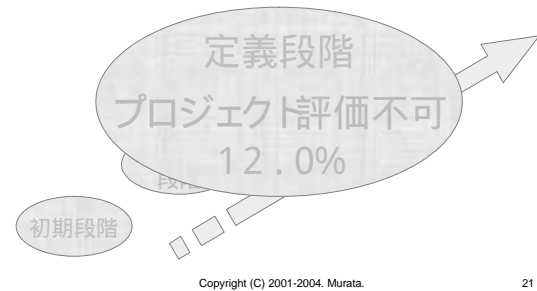
ソフトウェア開発の成熟度 レベル1



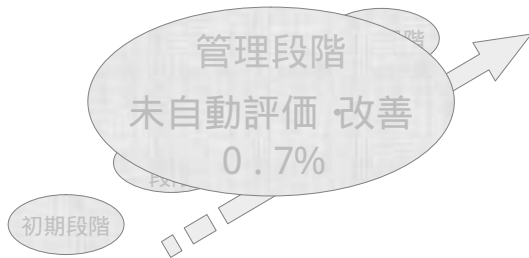
ソフトウェア開発の成熟度 レベル2



ソフトウェア開発の成熟度 レベル3



ソフトウェア開発の成熟度 レベル4



Copyright (C) 2001-2004. Murata.

22

ソフトウェア開発の成熟度 レベル5



Copyright (C) 2001-2004. Murata.

23

今期の内容

- | | |
|---------------------------------------|--------------------|
| 1. オープンソースのすすめ | 9/29 |
| 2. 変数 | 10/26 |
| 3. データ構造とアルゴリズム | 11/10, 17 |
| 4. 関数 | 11/24, 12/1 |
| 5. モジュール化 | 12/8, 11 |
| 6. ソフトウェアエンジニアリング
グリッドコンピューティングと応用 | 12/15, 22
12/22 |
| 7. 構造化手法 | 1/12 |
| 8. オブジェクト指向 | 1/12 |
| 9. ソフトウェアの品質 | 12/22 |
| 10. 保守しやすいソフトウェア | 12/22, 1/12 |

Copyright (C) 2001-2004. Murata.

24
