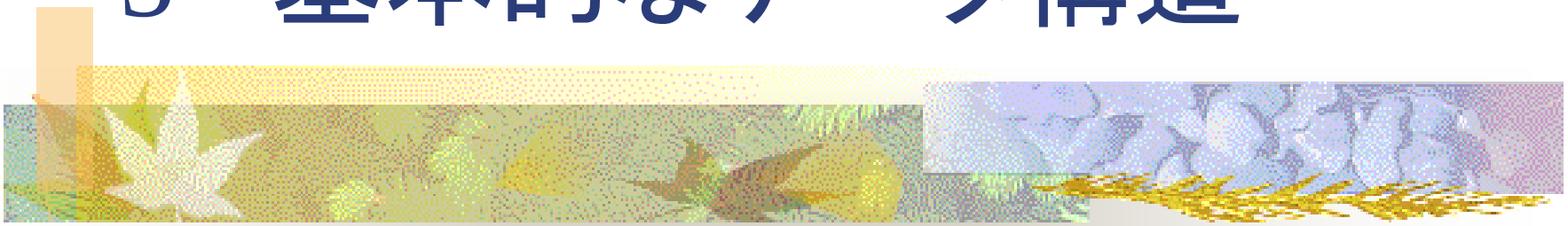
アルゴリズムとデータ構造a
5 - 基本的なデータ構造



大見　嘉弘

今日の授業

- 基本的なデータ構造
  - 単純型
  - ポインタ型
  - 配列型
  - レコード型

単純型

- もっとも基本的
- 1つの変数は1つのデータを持つ
- データの種類によって色々な型がある
  - 整数型
  - 実数型
  - 文字型
  - 論理型
  - 列挙型
  - …
- Javaでは基本データ型(Abstract Data Type)が相当する

単純型の種類 (※ カッコ内はJavaの型)

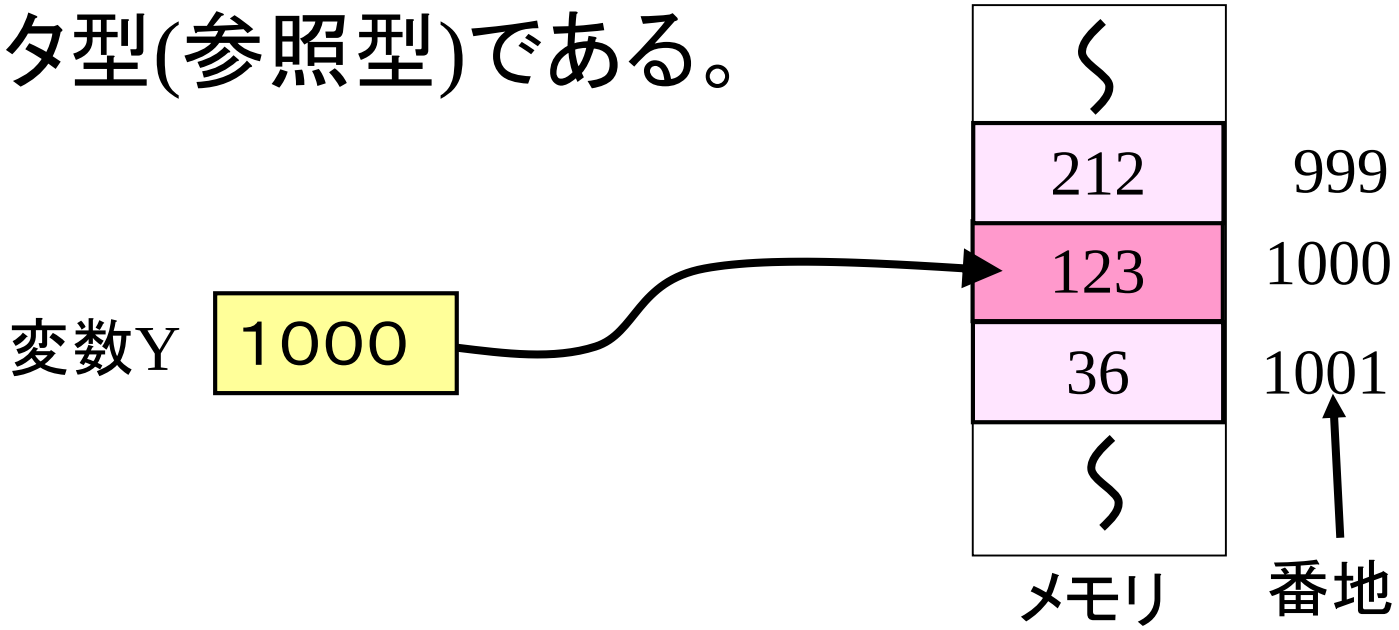
- 整数型 (int, long, byte)
  扱える範囲に下限、上限がある。
  例：byte は –128～+127,intは -2^{31}～$+2^{31}-1$
  ※ 言語によっては上限、下限がないものもある(可変長整数型)。
- 実数型(float, double)
  最近の言語は浮動小数点数型が主流。
  固定小数点型はあまり使われていない。

単純型の種類(2) (※ カッコ内はJavaの型)

- 文字型(char)
  - 英字、数字、記号等を文字データとして扱う。
  - 一文字の単位(言語により対応が違う)
    - 1バイト単位 (C言語のchar)
    - 多バイト文字 (Javaのchar, C言語のwchar)
  - 文字列が単純型かどうかは言語による
    - Javaの文字列(String)はオブジェクトの一種
- 論理型(boolean)　true, false の2値
- 列挙型(enum ※ Java 1.5よりサポート)
  - 変数が取りうる値をすべて列挙して定義。

ポインタ型

- データがどこにあるのかを示す。
- つまり、存在する場所(メモリ上の番地)を持つ。
- Javaの場合、オブジェクト型はすべてポインタ型(参照型)である。

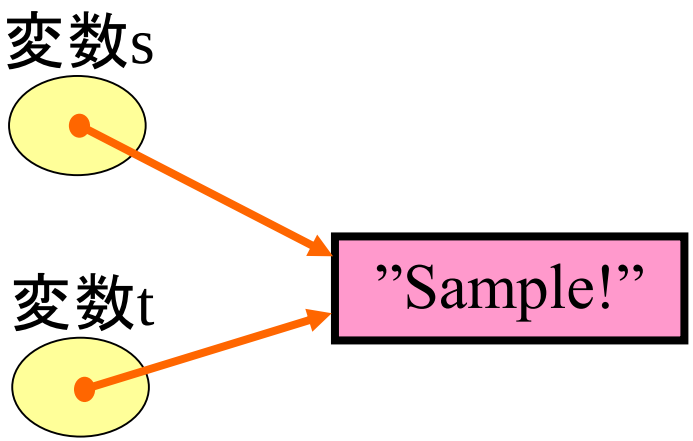


Javaの参照型の特徴

```
StringBuffer s = ”Sample”;
StringBuffer t = s;
t.append(”!”);
System.out.println(”s =” + s);
System.out.println(”t =” + t);
```

変数sとtは同じデータを指している。

そのデータに”!”を追加したので、sもtのどちらの内容も ”Sample!”。



ポインタ型の特徴

- データの大きさ(使用するバイト数)が巨大な場合、代入操作の効率が良い(データ自体をコピーしなくて良いため)。
- 1つのデータを複数の変数で表現できる(データの共有ができる)。
- C言語のようにポインタの値(番地)を直接扱うとプログラムの理解が難しく、重大なバグを引き起こす可能性が高くなる。Java言語ではポインタの値は隠されている。

配列型

- 配列は複数の要素から成る
- 各要素のデータ型は全て同じ
- １次元配列、２次元配列、３次元配列…
- ２次元配列以上を多次元配列という

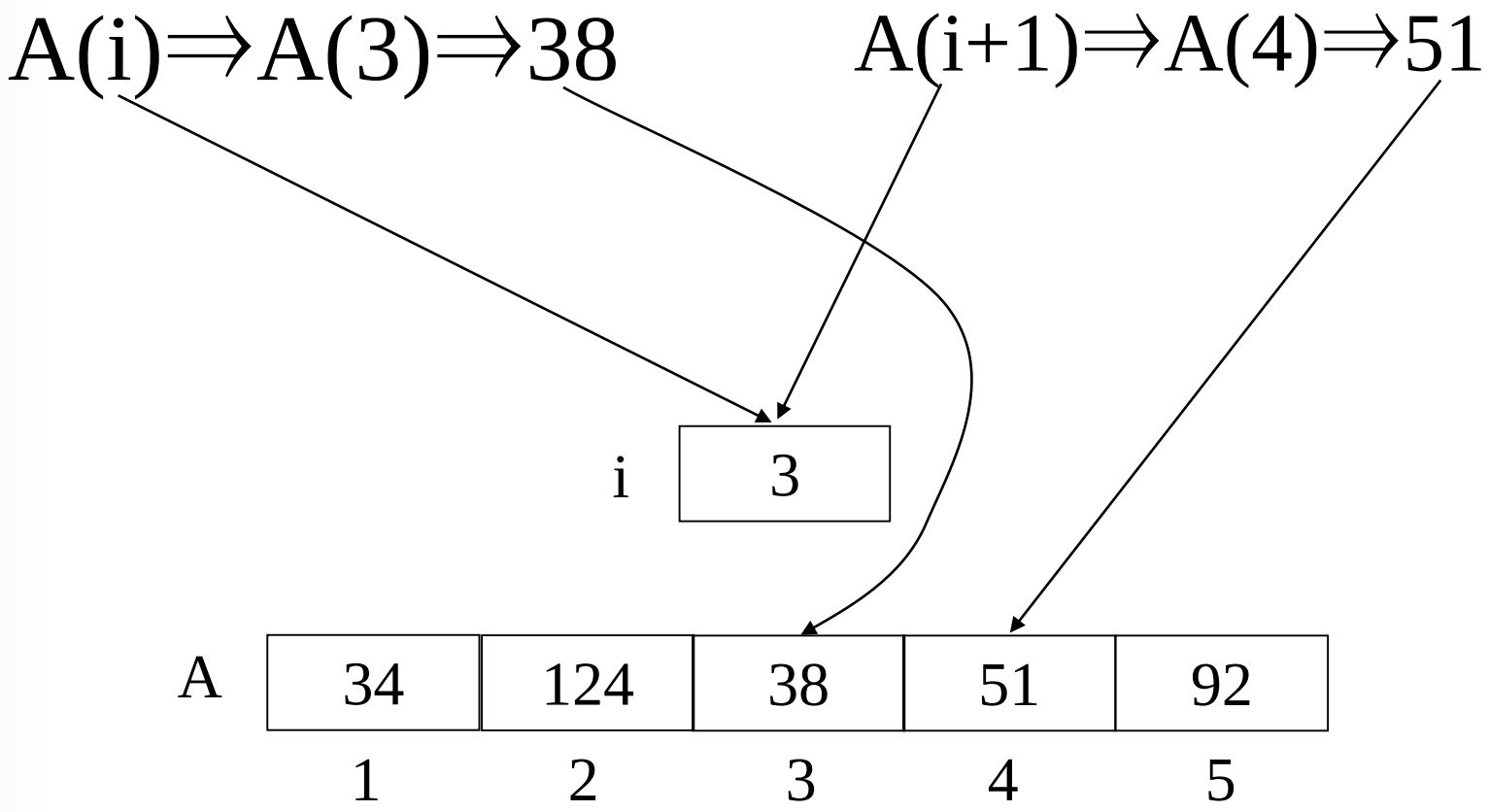
一次元配列

例：整数型の配列A(大きさ5)　　どの要素も整数型

A(1)←34
A(2)←124
A(3)←38
A(4)←51
A(5)←92

A	34	124	38	51	92
	1	2	3	4	5

一次元配列の例



二次元配列

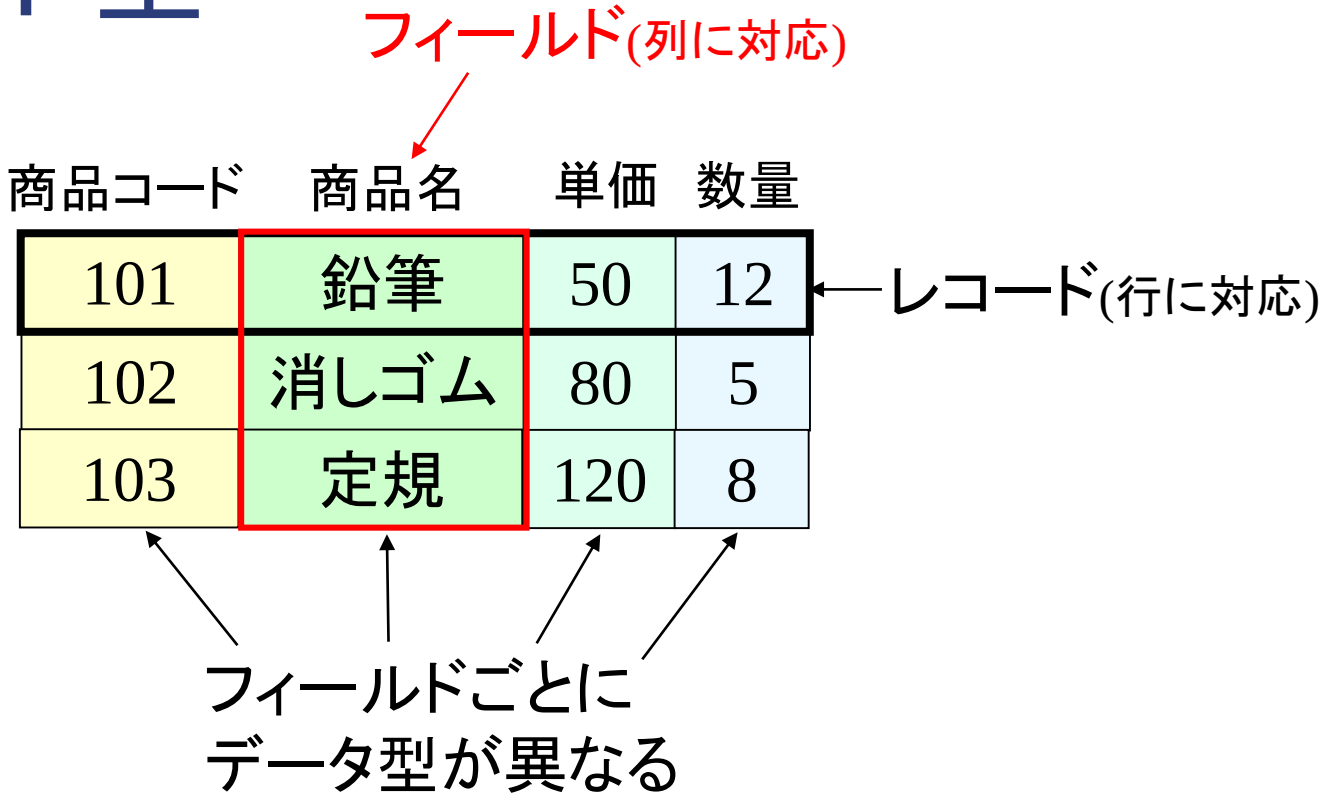
	1列目	2列目	3列目	4列目
1行目	A(1,1)	A(1,2)	A(1,3)	A(1,4)
2行目	A(2,1)	A(2,2)	A(2,3)	A(2,4)
3行目	A(3,1)	A(3,2)	A(3,3)	A(3,4)

A(行の添字,列の添字)

Javaの場合

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]

レコード型



RDB(リレーショナル型データベース)の構造

Javaでのレコード型の表現

1. レコードをクラスで定義
2. そのオブジェクトの配列を作る

```
class Record {
        int id;
        String name;
        int price;
        int quantity;
}
```

Recordクラス(1レコードに対応)を定義

レコードの内容は、商品コード(整数型)、商品名(文字列)、単価(整数型)、数量(整数型)

```
Record orders[] = new Record[100];
```

Record型の配列を作成する

宿題

- 配列Gに以下のような値が入っている時に、以下の処理を行った後の配列の状態を図で示せ。

配列G	100	200	300	400	500

  - G(5)←G(3)
  - G(5)←G(2)
  - G(3)←G(5)