コンピュータ・サイエンス2

第13回
プログラミング・アルゴリズム(実習)(続き)

人間科学科コミュニケーション専攻
白銀 純子

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

第13回の内容

- プログラミング・アルゴリズムの実習(続き)

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

プログラミング言語の復習

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

プログラミング言語の変遷[1](p. 76)

- プログラム: コンピュータに命令を伝えるための文書
- プログラミング言語: プログラムを記述するための言葉

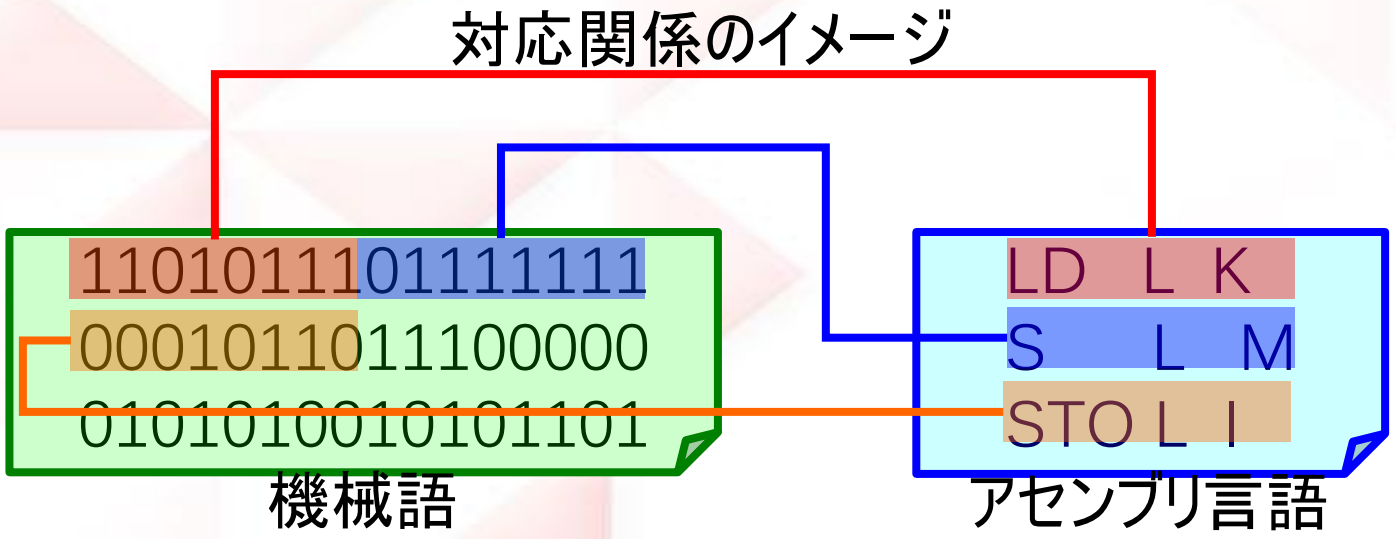
初期: 機械語でプログラムを記述

- 機械語: 0と1の2進数の形式の言葉
  - 不便!
  - アセンブリ言語が登場

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

アセンブリ言語

- 英語に似せた言語
- 機械語と1対1で対応
- アセンブリ言語のプログラムを機械語に翻訳
  - 翻訳ソフトウェア: アセンブラ



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

プログラミング言語の変遷[2](p. 76)

- 1954年にFORTRAN(FORmula TRANSlator)
  - IBM社が科学技術用言語として提唱
- 1959年にCOBOL(Common Business Oriented Language)
  - アメリカ国防省が商用言語として提唱
- 1962年にBASIC(Beginner's All purpose Symbolic Instruction Code)
  - ダートマス大学で初心者でも使える言語として提唱
  - ビル・ゲイツがよく利用し、Microsoft社が開発に注力

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

プログラミング言語の変遷[3](p. 76)

- 1972年にC言語
  - ベル研究所がOSなどの基幹ソフトウェアの開発用言語として開発
  - UNIXがOSとして初めてC言語で記述
- 1972年にSmalltalk
  - ゼロックス社のパロアルト研究所でオブジェクト指向言語として開発
- 1995年にJava
  - サン・マイクロシステムズ社(現オラクル社)でネットワーク対応言語として開発

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

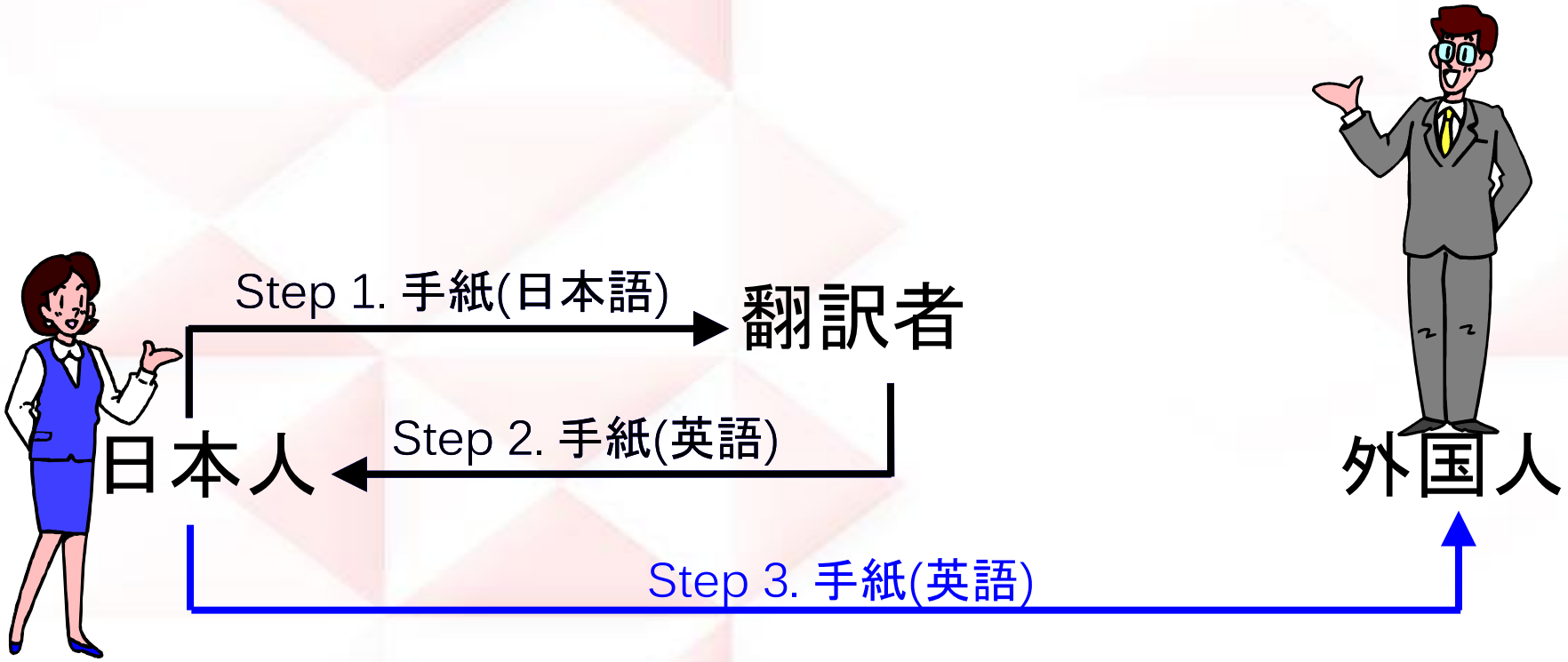
変換方式

- プログラミング言語で書かれた命令書: 機械語に変換しなければ、コンピュータは実行不可能
- 変換方式は大きく分けて2種類
  - コンパイラ型: 翻訳
  - インタプリタ型: 通訳

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

コンパイラ[概要](p. 78)

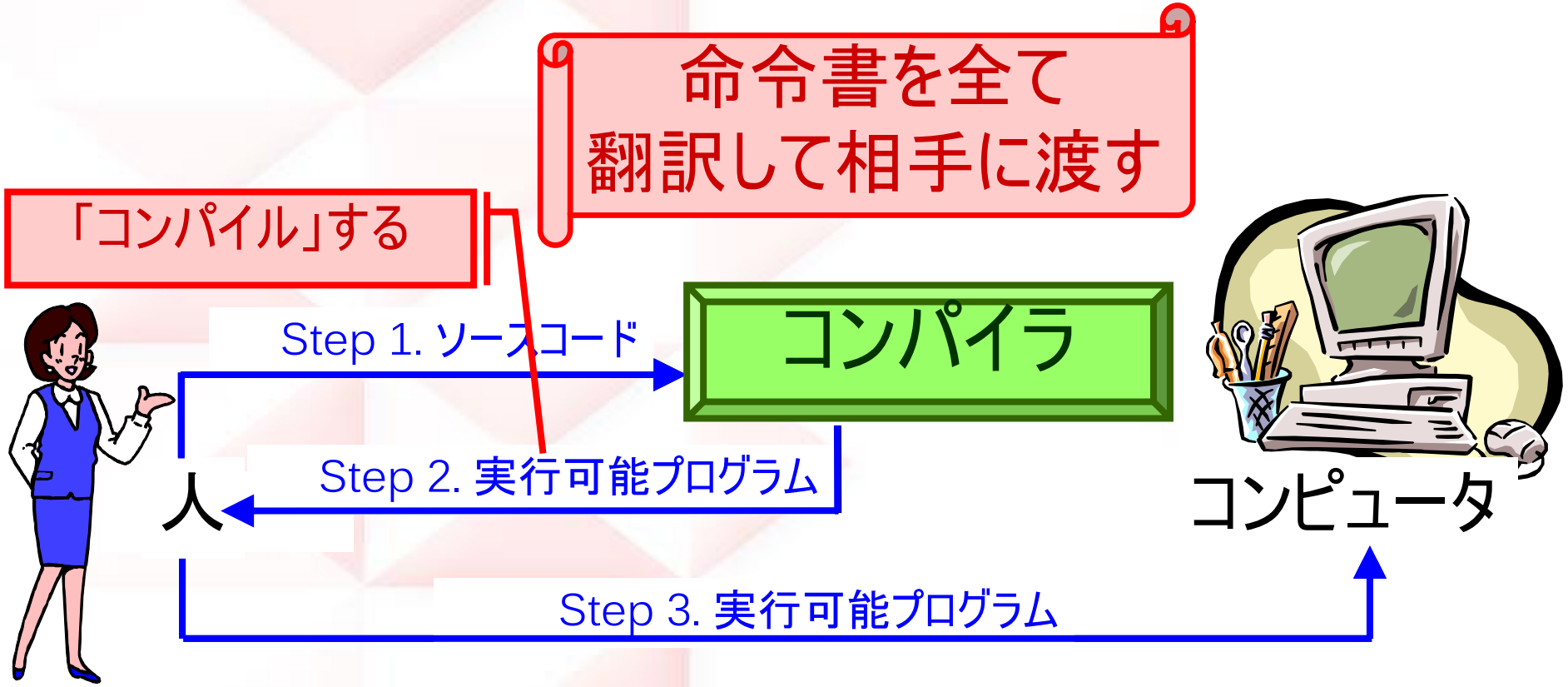
▶コンパイラ: 命令書を機械語に翻訳し、コンピュータで実行可能にするためのソフトウェア



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

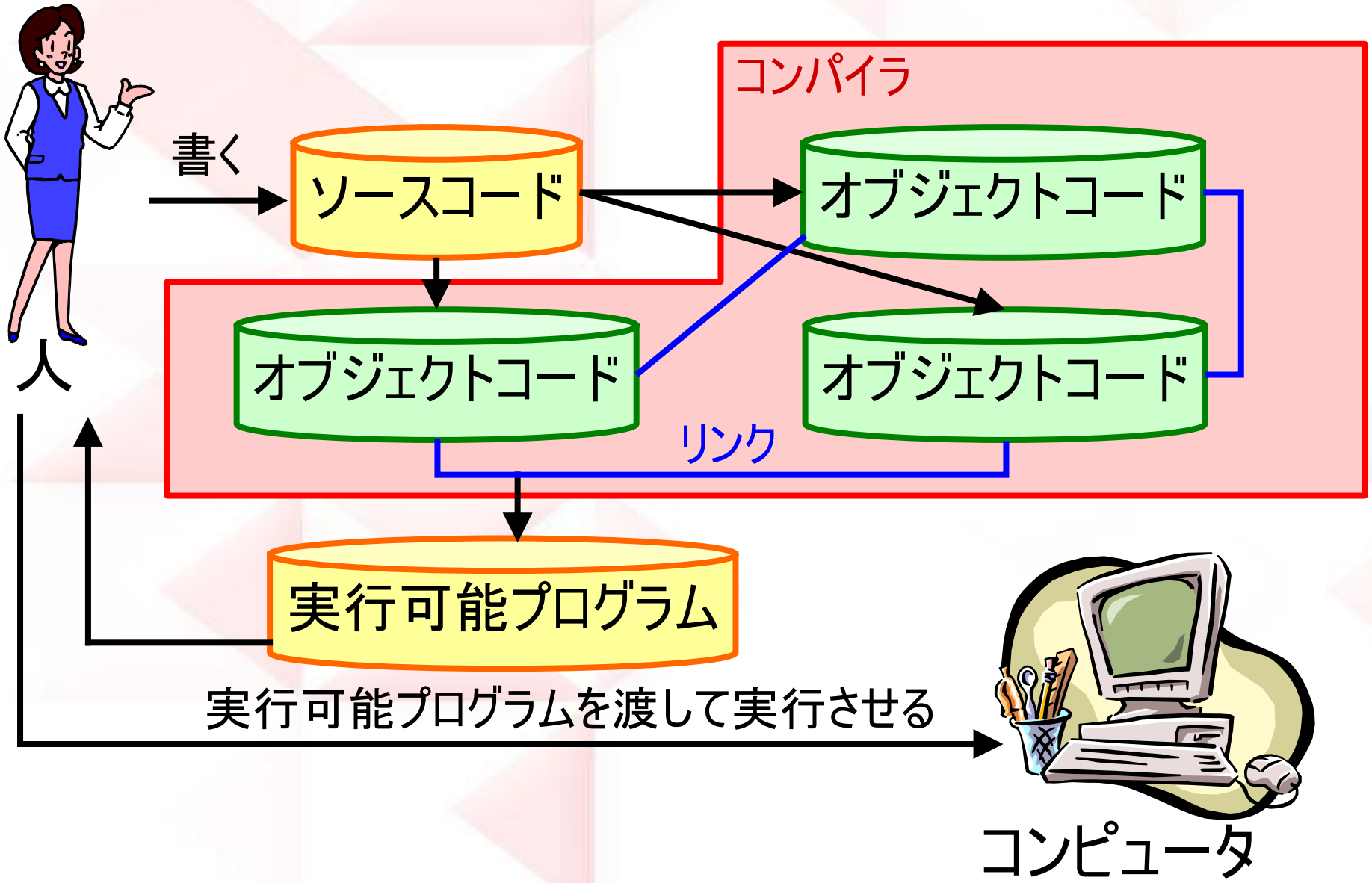
コンパイラ[概要](p. 78)

▶コンパイラ: 命令書を機械語に翻訳し、コンピュータで実行可能にするためのソフトウェア



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

コンパイラ[詳しく](p. 78)



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

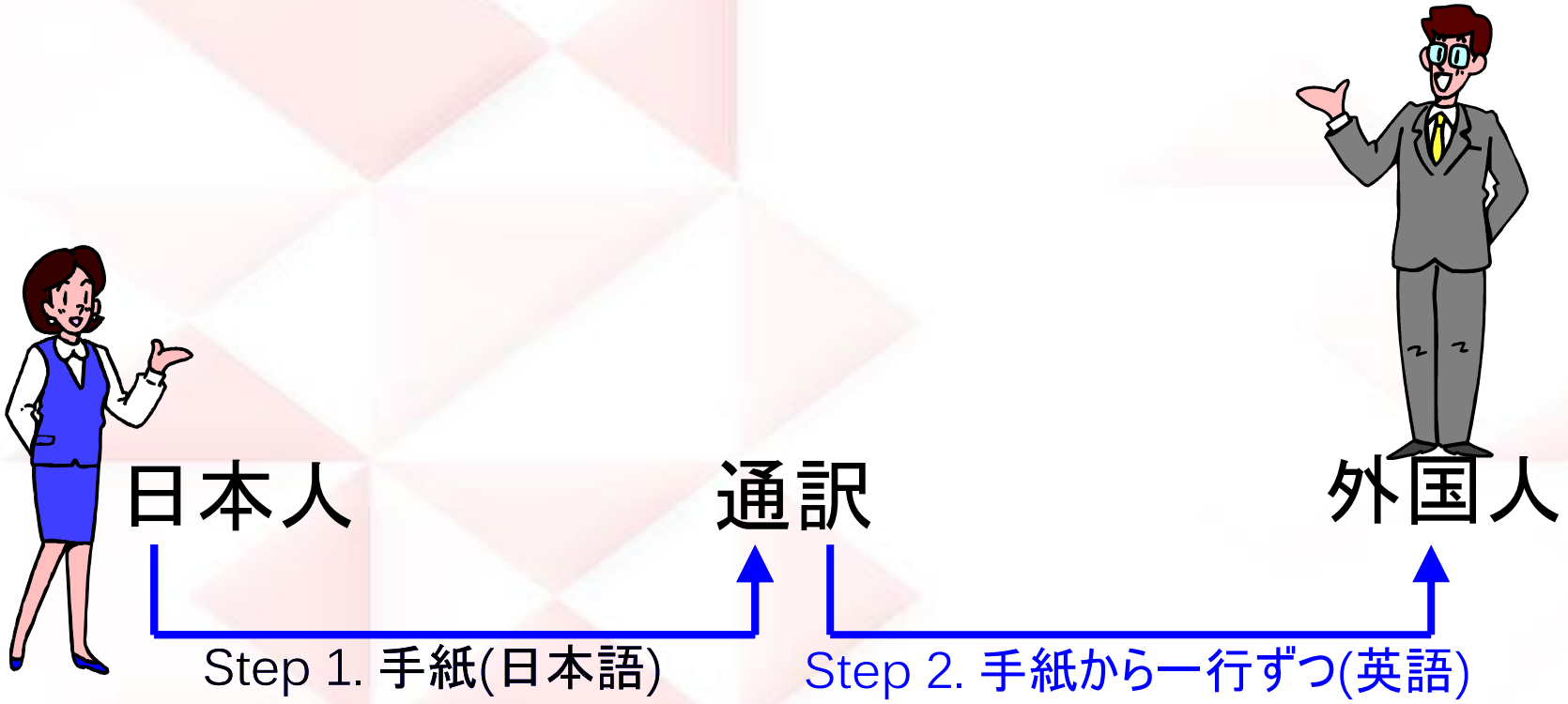
用語[1](p. 78)

- ソースコード: プログラミング言語で記述した命令書
- オブジェクトコード: ソースコードを翻訳したもの
- 実行可能プログラム: オブジェクトコードを連携させて、動作可能な形にしたもの
- プログラム: ソースコードと実行可能プログラムの双方の意味
  - どちらの意味で使われるかは、その時々で異なる

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

インタプリタ

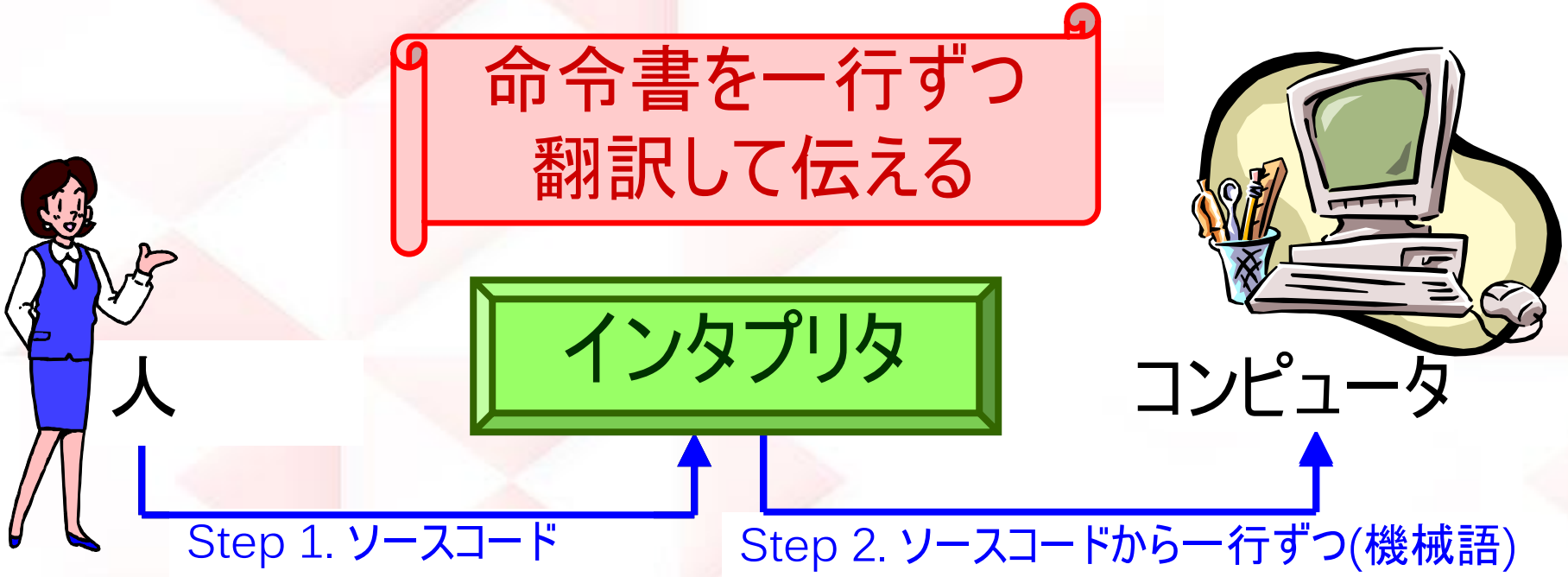
▶インタプリタ: 命令書を最初から1行ずつ読んで機械語に通訳するためのソフトウェア



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

インタプリタ

▶インタプリタ: 命令書を最初から1行ずつ読んで機械語に通訳するためのソフトウェア



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

アルゴリズムの復習

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

整列(ソート)アルゴリズム(p.123)

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

整列[ソート](p. 123)

- ソート: 複数の数を小さい順or大きい順に並べること
  - 選択ソート
  - バブルソート
  - 挿入ソート
  - クイックソート
  - etc.

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

バブルソート[1](p. 124)

- 前から2つずつ、数の大きさを比較して、小さい数を後ろに送っていく
  - 最後まで調べると、最も小さな数が一番後ろにある
  - この作業を、並べ替える数の個数だけ繰り返すと、数が大きい順に並ぶ

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

バブルソート[2](p. 124)

比べる

4 8 2 5

4より8の方が大きいので交換

比べる

8 4 2 5

2より4の方が大きいのでそのまま

比べる

8 4 2 5

2より5の方が大きいので交換

8 4 5 2

- 最も小さな数が一番後ろに来る
- 次は、一番後ろの1つ前まで(8, 4, 5)で同じようにする

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

ソート演習

- 配ったカードを、順番をバラバラにして…
  1. バブルソートで昇順に並べ替え(練習)
  2. バブルソートで昇順に並べ替え、かかった時間を計測(本番)
  3. 自分のやりやすい方法で昇順に並べ替え、かかった時間を計測
  4. 2. と3. の時間を比較(どちらが時間がかかっているか??)

本日の出席課題への報告のため、
2. と3. の時間はきちんとメモしておこう!

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

プログラミング実習

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

実習内容

- プログラミング実習
  - 命令書(プログラム)とはどのようなものか?
  - どうやって翻訳・通訳するか?
  - どうやって実行するか?
- アルゴリズム実習
  - アルゴリズムによって、そんなに処理時間に違いがあるか?

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

準備

- 授業のページから4つのファイルをダウンロード
  - BubbleSort.c
    - バブルソートをするC言語のプログラム
  - BubbleSort.html
    - バブルソートをするJavaScriptのプログラム
  - MergeSort.c
    - 併合ソートをするC言語のプログラム
  - MergeSort.html
    - 併合ソートをするJavaScriptのプログラム

※リンクをクリックするのではなく、右クリック→「ファイルをダウンロード」でダウンロードすること

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

C言語とJavaScript

- C言語
  - プログラミング言語の一種
  - 記述された命令書を機械語に翻訳した命令書を作成する形式の言語(コンパイラ型の言語)
- JavaScript
  - プログラミング言語の一種
  - 記述された命令書を機械語に通訳する形式の言語(インタプリタ型の言語)
  - Webページでよく利用

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.c[前半1]

```
#include<stdio.h>
#include<time.h>
int main() {
  int i, j, temp, num = 30000;
  int random[num];
  for (i = 0; i < num; i++) {
    random[i] = rand();
  }
  FILE *randFile = fopen("BubbleRandomNum.txt", "w");
  for (i = 0; i < num; i++) {
    fprintf(randFile, "%d¥n", random[i]);
  }
  fclose(randFile);

  int start, end, procTime;
  start = clock();
```

並べ替えをする
数の個数

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.c[前半2]

```
#include<stdio.h>
#include<time.h>
int main() {
  int i, j, temp, num = 30000;
  int random[num];
  for (i = 0; i < num; i++) {
    random[i] = rand();
  }
  FILE *randFile = fopen("BubbleRandomNum.txt", "w");
  for (i = 0; i < num; i++) {
    fprintf(randFile, "%d¥n", random[i]);
  }
  fclose(randFile);

  int start, end, procTime;
  start = clock();
```

並べ替えをする数を
自動的に作成

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.c[前半3]

```
#include<stdio.h>
#include<time.h>
int main() {
  int i, j, tem
  int rando
  for (i = 0;
    random[i] = rand();
  }
  FILE *randFile = fopen("BubbleRandomNum.txt", "w");
  for (i = 0; i < num; i++) {
    fprintf(randFile, "%d¥n", random[i]);
  }
  fclose(randFile);

  int start, end, procTime;
  start = clock();
```

自動的に作った数を
「BubbleRandomNum.txt」ファイルに書き出し

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.c[後半1]

```
for (i = 0; i < num; i++) {
  for (j = 0; j < num - i - 1; j++) {
    if (random[j] > random[j + 1]) {
      temp = random[j + 1];
      random[j + 1] = random[j];
      random[j] = temp;
    }
  }
}
end = clock();
procTime = end - start;
printf("Time(Bubble Sort of %d Numbers): %lf second¥n", num,
        (double) procTi

FILE *bubbleFile = fopen("BubbleSortNum.txt", "w");
for (i = 0; i < num; i++) {
  fprintf(bubbleFile, "%d¥n", random[i]);
}
fclose(bubbleFile);
}
```

バブルソートをする処理部分

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.c[後半2]

```
for (i = 0; i < num; i++) {
  for (j = 0; j < num - i - 1; j++) {
    if (random[j] > random[j + 1]) {
      temp = random[j + 1];
      random[j + 1] = random[j];
      random[j] = temp;
    }
  }
}
end = clock();
procTime = en
printf("Time(Bu
        (double) procTime / CLOCKS_PER_SEC);

FILE *bubbleFile = fopen("BubbleSortNum.txt", "w");
for (i = 0; i < num; i++) {
  fprintf(bubbleFile, "%d¥n", random[i]);
}
fclose(bubbleFile);
}
```

数を並べ替えた結果を
「BubbleSortNum.txt」ファイルに書き出し

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.html[前編1]

```
<html>
<head>
<title>バブルソート</title
<script language="Ja
NUM = 5000;
random = new Array();
for (i=0; i<=NUM; i++) {
random[i] = Math.floor(NUM *
Math.random());
}
function bubbleSort() {
  for (i = 0; i < NUM; i++) {
    for (j = 0; j < NUM - i - 1; j++) {
      if (random[j] > random[j + 1]) {
        temp = random[j + 1];
        random[j + 1] = random[j];
        random[j] = temp;
      }
    }
  }
}
```

並べ替えをする
数の個数

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.html[前編2]

```
<html>
<head>
<title>バブルソート</title>
<script language="JavaScript"><!--
NUM = 5000;
random = new Array();
for (i=0; i<=NUM; i++) {
random[i] = Math.floor(NUM * Math.random());
}
function bubbleSort() {
  for (i = 0; i < NUM; i++) {
    for (j = 0; j < NUM - i - 1; j++) {
      if (random[j] > random[j + 1]) {
        temp = random[j + 1];
        random[j + 1] = random[j];
        random[j] = temp;
      }
    }
  }
}
```

並べ替えをする数を
自動的に作成

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.html[前編3]

```
<html>
<head>
<title>バブルソート</title>
<script language="JavaScript"><!--
NUM = 5000;
random = new Array();
for (i=0; i<=NUM; i++) {
random[i] = Math.floor(NUM*Math.random())
}
function bubbleSort() {
 for (i = 0; i < NUM; i++) {
  for (j = 0; j < NUM - i - 1; j++) {
   if (random[j] > random[j + 1]) {
    temp = random[j + 1];
    random[j + 1] = random[j];
    random[j] = temp;
   }
  }
 }
}
```

バブルソートをする処理部分

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.html[中編]

```
function writeNumber() {
 for (i=0; i<NUM; i++) document.write(random[i]," ");
 document.write("<br>");
}
// --></script>
</head>
<body>
<h1>バブルソート</h1>
<h2>並び替え前</h2>
<script langauge="JavaScript"><!--
writeNumber();
// --></script>
```

並べ替えた数をブラウザに
表示する部分

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.html[後編]

```
<h2>並び替え後</h2>
<script langauge="JavaScript"><!--
Start = new Date();
bubbleSort();
Stop = new Date();
startTime = Start.getTime();
stopTime = Stop.getTime();
resultTime = stopTime - startTime;
writeNumber();
document.write("<br>かかった時間: " + resultTime + "msec<br>");
// --></script>
</body>
</html>
```

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.c[前編1]

```
#include<stdio.h>

int num = 30000;
void mergeSort(int random[], int start, int end) {
  int middle, i, j, k;
  int temp[num];
  if (start >= end) {
    return;
  }
  middle = (start + end) / 2;
  mergeSort(random, start, middle);
  mergeSort(random, middle + 1, end);

  for (i = start; i <= middle; i++) {
    temp[i] = random[i];
  }
  for (i = middle + 1, j = end; i <= end; i++, j--) {
    temp[i] = random[j];
  }
  i = start;
  j = end;
```

並べ替えをする
数の個数

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.c[前編2]

```
#include<stdio.h>

int num = 30000;
void mergeSort(int random[], int start, int end) {
  int middle, i, j, k;
  int temp[num];
  if (start >= end) {
    return;
  }
  middle = (start + end) / 2;
  mergeSort(random, start, middle);
  mergeSort(random, middle + 1, end);

  for (i = start; i <= middle; i++) {
    temp[i] = random[i];
  }
  for (i = middle + 1, j = end; i <= end; i++, j--) {
    temp[i] = random[j];
  }
  i = start;
  j = end;
```

併合ソートの処理部分(前半)

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.c[中編1]

```
  for (k = start; k < end; k++) {
    if (temp[i] <= temp[j]) {
      random[k] = temp[i++];
    } else {
      random[k] = temp[j--];
    }
  }
}
int main() {
  int i, j, temp;
  int random[num];
  for (i = 0; i < num; i++) {
    random[i] = rand();
  }
  FILE *randFile = fopen("MergeRandomNum.txt", "w");

  for (i = 0; i < num; i++) {
    fprintf(randFile, "%d¥n", random[i]);
  }
  fclose(randFile);
```

併合ソートの処理部分(後半)

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.c[中編2]

```
  for (k = start; k < end; k++) {
    if (temp[i] <= temp[j]) {
      random[k] = temp[i++];
    } else {
      random[k] = temp[j--];
    }
  }
}
int main() {
  int i, j, temp;
  int random[num];
  for (i = 0; i < num; i++) {
    random[i] = rand();
  }
  FILE *randFile = fopen("MergeRandomNum.txt", "w");

  for (i = 0; i < num; i++) {
    fprintf(randFile, "%d¥n", random[i]);
  }
  fclose(randFile);
```

並べ替えをする数を
自動的に作成

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.c[中編3]

```
  for (k = start; k < end; k++) {
    if (temp[i] <= temp[j]) {
      random[k] = temp[i++];
    } else {
      random[k] = temp[j--];
    }
  }
}
int main() {
  int i, j, te
  int rand
  for (i = 0
    random[i] = rand();
  }
  FILE *randFile = fopen("MergeRandomNum.txt", "w");

  for (i = 0; i < num; i++) {
    fprintf(randFile, "%d¥n", random[i]);
  }
  fclose(randFile);
```

自動的に作った数を
「MergeRandomNum.txt」ファイルに書き出し

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.c[後編]

```
  int start, end, procTime;
  start = clock();
  mergeSort(random, 0, num - 1);
  end = clock();

  procTime = end - start;
  printf("Time(Merge Sort of %d Numbers): %f second¥n", num,
          (double) procTime / CLOCKS_PER_SEC);
  FILE *mergeFile = fopen("MergeSortNum.txt", "w");
  for (i = 0; i < num; i++) {
    fprintf(mergeFile, "%d¥n", random[i]);
  }
  fclose(mergeFile);
}
```

数を並べ替えた結果を
「MergeSortNum.txt」ファイルに書き出し

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.html[前編1]

```
<html>
<head>
<title>併合ソート</title>
<script language="Ja
NUM = 5000;
random = new Array();
for (i = 0; i <= NUM; i++) {
random[i] = Math.floor(NUM * Math.random());
}
temp = new Array();
function mergeSort(start,end) {
    if (start >= end) return;
        var middle = Math.floor((start + end) / 2);
        mergeSort(start, middle);
        mergeSort(middle + 1, end);
        var p = 0;
        for (i = start; i <= middle; i++) temp[p++] = random[i];
            var i = middle + 1;
            var j = 0;
            var k = start;
```

並べ替えをする
数の個数

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.html[前編2]

```
<html>
<head>
<title>併合ソート</title>
<script language="JavaScript"><!--
NUM = 5000;
random = new Array();
for (i = 0; i <= NUM; i++) {
random[i] = Math.floor(NUM * Math.random());
}
temp = new Array();
function mergeSort(start,end) {
    if (start >= end) return;
        var middle = Math.floor((start + end) / 2);
        mergeSort(start, middle);
        mergeSort(middle + 1, end);
        var p = 0;
        for (i = start; i <= middle; i++) temp[p++] = random[i];
            var i = middle + 1;
            var j = 0;
            var k = start;
```

並べ替えをする数を
自動的に作成

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.html[前編3]

```
<html>
<head>
<title>併合ソート</title>
<script language="JavaScript"><!--
NUM = 5000;
random = new Array();
for (i = 0; i <= NUM; i++) {
random[i] = Math.floor(NUM * Math.rando
}
temp = new Array();
function mergeSort(start,end) {
  if (start >= end) return;
    var middle = Math.floor((start + end) / 2);
    mergeSort(start, middle);
    mergeSort(middle + 1, end);
    var p = 0;
    for (i = start; i <= middle; i++) temp[p++] = random[i];
      var i = middle + 1;
      var j = 0;
      var k = start;
```

併合ソートの処理部分(前半)

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.html[中編1]

```
            while ((i <= end) && (j < p)) {
                if (temp[j] <= random[i])  {
                    random[k++] = temp[j++];
                } else {
                    random[k++] = random[i++];
                }
            }
            while (j < p) {
                random[k++] = temp[j++];
            }
}
function writeNumber() {
    for (i = 0; i < random.length; i++) {
        document.write(random[i]," ");
    }
    document.write("<br>");
}
// --></script>
</head>
```

併合ソートの処理部分(後半)

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.html[中編2]

```
            while ((i <= end) && (j < p)) {
                if (temp[j] <= random[i])  {
                    random[k++] = temp[j++];
                } else {
                    random[k++] = random[i++];
                }
            }
            while (j < p) {
                random[k++] = temp[j+
            }
}
function writeNumber() {
    for (i = 0; i < random.length; i++) {
        document.write(random[i]," ");
    }
    document.write("<br>");
}
// --></script>
</head>
```

並べ替えた数をブラウザに
表示する部分

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

MergeSort.html[後編]

```
<body>
<h1>併合ソート</h1>
<h2>並び替え前</h2>
<script langauge="JavaScript"><!--
writeNumber();
// --></script>

<h2>並び替え後</h2>
<script langauge="JavaScript"><!--
Start = new Date();
mergeSort(0,random.length-1);
Stop = new Date();
startTime = Start.getTime();
stopTime = Stop.getTime();
resultTime = stopTime - startTime;
writeNumber();
document.write("<br>かかった時間: " + resultTime + "msec<br>");
// --></script>
</body>
</html>
```

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

ファイルの役割

- BubbleSort.c
  - 「BubbleRandomNum.txt」に、並び替え前の数を保存
  - 「BubbleSortNum.txt」に、並び替え後の数を保存
- BubbleSort.html
  - 並び替え前と後の数をブラウザに表示
- MergeSort.c
  - 「MergeRandomNum.txt」に、並び替え前の数を保存
  - 「MergeSortNum.txt」に、並び替え後の数を保存
- MergeSort.html
  - 並び替え前と後の数をブラウザに表示

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.c, MergeSort.c[1]

- コンパイルして機械語のファイルを作成

1. Finder→「アプリケーション」→「ユーティリティ」→「ターミナル」をダブルクリック
2. それぞれのファイルをコンパイル
   - BubbleSort.cの場合:
     gcc -o BubbleSort BubbleSort.c
     と入力し、「Return」キーを押す
   - MergeSort.cの場合:
     gcc -o MergeSort MergeSort.c
     と入力し、「Return」キーを押す

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.c, MergeSort.c[2]

3. Finderで、「BubbleSort」ファイルと「MergeSort」ファイルができていることを確認
   - ▶BubbleSort: BubbleSort.cを機械語に翻訳したファイル
   - ▶MergeSort: MergeSort.cを機械語に翻訳したファイル

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.c, MergeSort.c[2]

- プログラミング言語が機械語に翻訳されているかを確認
  - BubbleSort.cをJeditで開いてみる
  - BubbleSortをJeditで開いてみる
  - MergeSort.cをJeditで開いてみる
  - MergeSortをJeditで開いてみる

- gcc: C言語のプログラムを機械語に翻訳するためのコンパイラ
  - ✓「gcc -o ファイル1 ファイル2」で、「ファイル2のプログラムを機械語に翻訳し、ファイルに保存する」という命令

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.c, MergeSort.c[3]

- プログラムを実行
  - BubbleSort.cの場合: ターミナルで「./BubbleSort」と入力し、「Return」キーを押す
  - MergeSort.cの場合: ターミナルで「./MergeSort」と入力し、「Return」キーを押す

    - BubbleRandomNum.txtとBubbleSortNum.txtを開き、数が並び替えされているかどうかを確認
    - MergeRandomNum.txtとMergeSortNum.txtを開き、数が並び替えされているかどうかを確認
    - 「Time: かかった時間 second」とターミナル上に表示されるので、かかった時間を比較

※かかった時間の単位は秒

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

BubbleSort.html, MergeSort.html

▶BubbleSort.htmlとMergeSort.htmlをWebブラウザで表示

- ブラウザで表示された数が、「並び替え前」と「並び替え後」で並び替えられているかどうかを確認
- 「かかった時間msec」(ブラウザの一番下に表示)ので、BubbleSort.htmlとMergeSort.htmlでかかった時間を比較
- BubbleSort.cとBubbleSort.htmlでかかった時間を比較(※BubbleSort.cは数が30000個、BubbleSort.htmlは数が5000個)
- MergeSort.cとMergeSort.htmlでかかった時間を比較(※MergeSort.cは数が30000個、MergeSort.htmlは数が5000個)

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

本日の出席課題

- 回答場所: 授業の資料のページからアクセス
- 回答内容
  - カードを使ってバブルソートでソートを行ったときに、それぞれかかった時間
  - カードを使って自分のやり方でソートを行ったときに、それぞれかかった時間
  - C言語のプログラムでバブルソートを行ったときの時間
  - C言語のプログラムで併合ソートを程度行ったときの時間
  - Webブラウザでバブルソートを程度行ったときの時間
  - Webブラウザで併合ソートを行ったときの時間

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

次回

- 補講: 1月21日(木) 3限 24101教室

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.