



# 序論：この講義の目的

オブジェクト指向プログラミング特論

2020年度

只木進一：理工学研究科

# オブジェクト指向 Object-Oriented

- オブジェクト指向開発・オブジェクト指向プログラミング
  - 身に付いたか?
  - 理解している?
- 問題をオブジェクトとその操作・運動としてモデル化できるか?
- クラスの継承は理解したか。使えるか。
- 抽象クラスは使えるか?

# プログラミングに必要なことは

- ➡ 講義で扱ったのは「文法」
- ➡ 全体の設計の技術が必要
  - ➡ 対象問題の整理
  - ➡ クラス設計
  - ➡ テスト
  - ➡ 改善

## この講義の目標

- 実用的な例を通じてオブジェクト指向プログラミングを身につける
- Javaらしいコーディング
- スマートなプログラミングの枠組み
  - 効率よく、良質なプログラムを書く
- プログラミングのスキルを向上させる

# プログラミング上達のために

- モジュール化
- 全体の構成を整理する
- データ・モデル、全体の流れ、UIを分離する
- 適切なライブラリの活用
- 良い例題の学習
- 良い教科書を持つ

# 今日のタスク

- 作業環境の整備
- サンプルプログラムの取得
- MergeSortを例にJavaの基本を復習
  - 再帰的MergeSortのアルゴリズム
  - 実際の動作の確認
  - 実装上の注意

# 準備

## ■ JDKとNetBeansの更新

■ <https://aws.amazon.com/jp/corretto/>

■ <https://netbeans.apache.org/download/index.html>

## ■ JDK APIマニュアル

■ <https://www.oracle.com/technetwork/jp/java/javase/documentation/api-jsp-316041-ja.html>

# サンプルプログラム

- ➡ <https://github.com/oop-mc-saga/Sort>
- ➡ example0/パッケージの中を見る



## Steps

1. **Remote Repository**
2. Remote Branches
3. Destination Directory

## Remote Repository

Specify Git Repository Location:

Repository URL:

http[s]://host.xz[:port]/path/to/repo.git/

User:

(leave blank for anonymous access)

Password:

☐ Save Password

[Proxy Configuration...](#)

Specify Destination Folder:

Clone into:

[Browse...](#)

(Leave empty to specify the destination later)

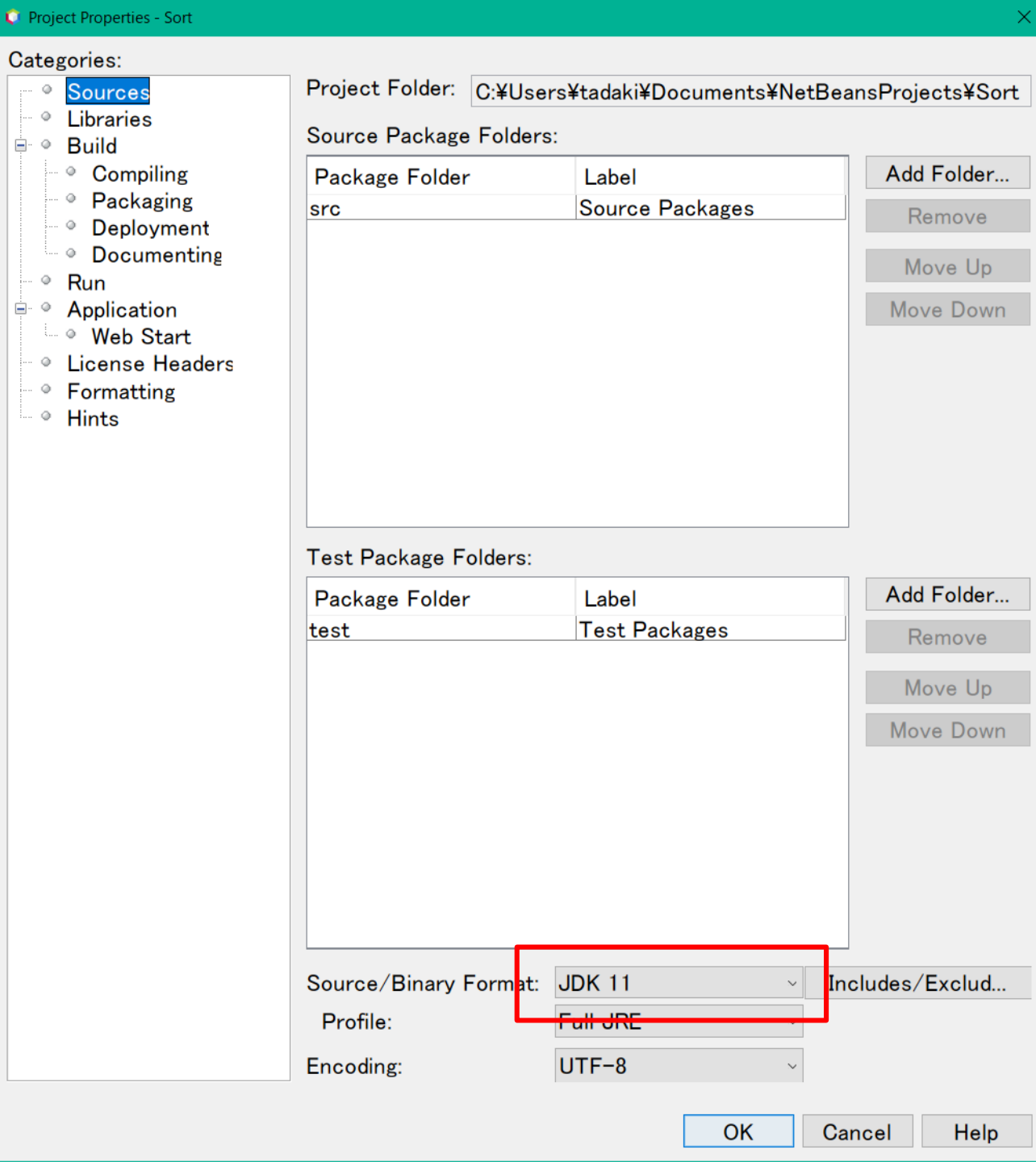
< Back

Next >

Finish

Cancel

Help



## 例 : Merge Sort アルゴリズム

```
sort(left, right){  
    if (right == left + 1) return  
    middle = (left + right)/2  
    sort(left, middle)  
    sort(middle, right)  
    merge(left, middle, right)  
}
```

# Merge Sort

## リストの分離

3	8	5	2	7	6	1	4
---	---	---	---	---	---	---	---

3	8	5	2
---	---	---	---

7	6	1	4
---	---	---	---

3	8
---	---

5	2
---	---

7	6
---	---

1	4
---	---

3	8	5	2
---	---	---	---

7	6	1	4
---	---	---	---

# Merge Sort

## リストの結合

3	8	5	2
---	---	---	---

7	6	1	4
---	---	---	---

3	8	2	5
---	---	---	---

6	7	1	4
---	---	---	---

2	3	5	8
---	---	---	---

1	4	6	7
---	---	---	---

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

# Merge Sort

## 再帰の実際

3	8	5	2	7	6	1	4
---	---	---	---	---	---	---	---

3	8	5	2	7	6	1	4
---	---	---	---	---	---	---	---

3	8	5	2	7	6	1	4
---	---	---	---	---	---	---	---

3	8	5	2	7	6	1	4
---	---	---	---	---	---	---	---

merge

3	8	5	2	7	6	1	4
---	---	---	---	---	---	---	---

3	8	5	2	7	6	1	4
---	---	---	---	---	---	---	---

3	8	5	2	7	6	1	4
---	---	---	---	---	---	---	---

3	8	5	2	7	6	1	4
---	---	---	---	---	---	---	---

merge

3	8	2	5	7	6	1	4
---	---	---	---	---	---	---	---

merge

2	3	5	8	7	6	1	4
---	---	---	---	---	---	---	---

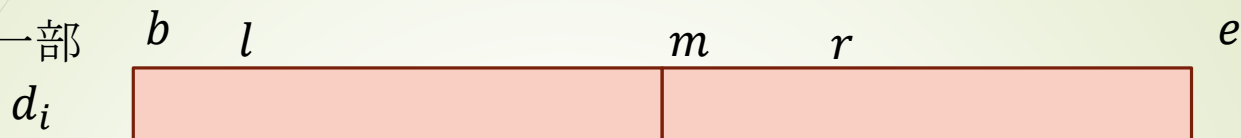
## 実装上の注意

- 元のリストの範囲を指定して書き換え
- mergeの際には、dummyリストへ書きこみ、最後に上書き



# 二つの整列済みリストの結合

元のリストの一部



```
List dummy;
while (  $l < m \vee r < e$  ){
  if (  $l \geq m$  ){//左側終了
    右側を詰め込む }
  if (  $r \geq e$  ){//右側終了
    左側を詰め込む}
  if (  $d_l < d_r$  ){ dummy.add( $d_l$ );  $l++$ ; }
  else { dymmy.add( $d_r$ );  $r++$ ;  $c++$ ; }
}
```

## 参考書

- Patrick Niemeyer, Jonathan Knudsen, *Learning Java 4th ed.* (O' Reilly, 2013).
- D. Poo, D. Kiong and S. Ashok, *Object-Oriented Programming and java* (Springer, 2008).
- Richard Warburton, *Java 8 Lambdas*, (O' Reilly, 2014).