Extracting superclass

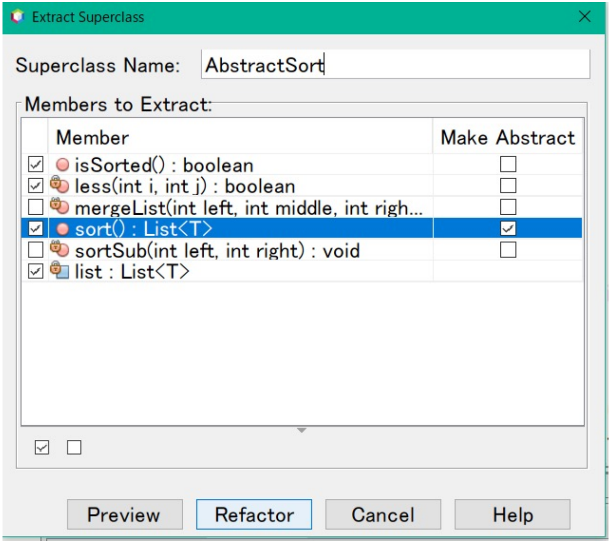
Object Oriented Programming
2022 First Semester
Shin-chi Tadaki (Saga University)

Extracting superclass

- Extract common features from existing classes
- Use the *refactoring* function in NetBeans
- Preparation
  - Copy to `example2`
    - `BubbleSort`
    - `MergeSort`
  - Delete `import example1.*`

Extract features from `MergeSort`

- Extract as the current form
  `less()`, `isSorted()`, `list`
- Extract as `abstract`
  `sort()`
- Save as `AbstractSort`
- Confirm the constructor

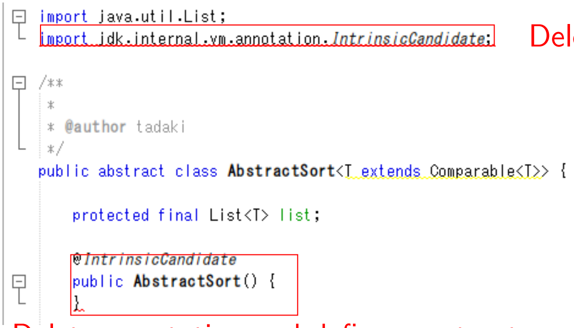

Extract Superclass

Superclass Name: AbstractSort

Members to Extract:

	Member	Make Abstract
✓	isSorted() : boolean	☐
✓	less(int i, int j) : boolean	☐
☐	mergeList(int left, int middle, int righ...	☐
✓	sort() : List<T>	✓
☐	sortSub(int left, int right) : void	☐
✓	list : List<T>	

Modify `AbstractSort`

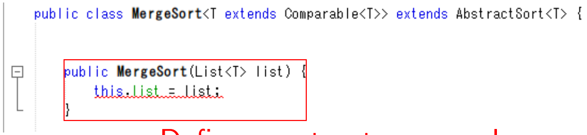


```
import java.util.List;
import jdk.internal.vm.annotation.IntrinsicCandidate;

/**
 *
 * @author tadaki
 */
public abstract class AbstractSort<T extends Comparable<T>> {

    protected final List<T> list;

    @IntrinsicCandidate
    public AbstractSort() {
    }
```

Delete

Delete annotation and define constructor properly

Modify `MergeSort`



```java
public class MergeSort<T extends Comparable<T>> extends AbstractSort<T> {

    public MergeSort(List<T> list) {
        this.list = list;
    }
```

Define constructor properly

Subclasses of `AbstractSort`

- `MergeSort`
- `BubbleSort`
- Subclasses override `sort()`

Exercise: Selection Sort

Algorithm 1 Selection Sort for list $d_i (0 \leq i < n)$

```
for $i = 0; i < n - 1; i++$ do
    $m = i$
    for $j = i + 1; j < n; j++$ do
        if $d_j < d_m$ then
            $m = j$
        end if
    end for
    if $m \neq i$ then
        swap($i, m$)
    end if
end for
```

Exercise

- Define `SelectionSort` class as a subclass of `AbstractSort`.
- Define `protected void swap(int,int)` in `AbstractSort`.
- And confirm it work.