Report#chap08

Due day: 2020.11.10 Tue

21600447 유승준

ECE20016 - 01 자바프로그래밍언어

**Problem#1:**

Modify the Student class in Listing 8.2 so that it implements the Comparable interface.

Define the compareTo method to order Student objects based on the value in studentNumber.

In a main method create an array of at least 5 Student objects, sort them using Arrays.sort, and output the students.

They should be listed by ascending student number. Next, modify the compareTo method so it orders Student objects based on the lexicographic ordering of the name variable.

Without modification to the main method, the program should now output the students ordered by name.

해석: Listing 8.2의 Student 클래스를 수정하여 Comparable 인터페이스를 구현합니다.

studentNumber의 값을 기준으로 Student 개체를 정렬하도록 compareTo 메서드를 정의한다.

기본 방법에서 최소 5개의 Student 개체의 배열을 만들고 Arrays.sort를 사용하여 정렬한 다음 학생을 출력한다.

학생 번호가 오름차순으로 나열되어야 한다. 다음으로, 이름 변수의 사전식 순서를 기반으로 Student 개체를 정렬하도록 compareTo 메서드를 수정한다.

메인 메서드를 수정하지 않고 프로그램은 이제 이름순으로 학생을 출력해야 한다.

**Concept:**

**#1.**

**- Access Modifier:**

1. **Private:** private 이 붙은 변수, 메소드는 해당 클래스에서만 접근이 가능하다.
2. **Public:** public 접근제어자가 붙은 변수, 메소드는 어떤 클래스에서라도 접근이 가능하다.

**- Static :** static은 ‘고정된’의 의미를 가지고 있다. Static이라는 키워드를 사용하여 Static변수와 Static메소드를 만들 수 있는데 다른 말로 정적필드와 정적 메소드라고도 하며 이 둘을 합쳐 정적 멤버라고 한다. (클래스 멤버라고도 한다.) 정적 필드와 정적 메소드는 객체(인스턴스)에 소속된 멤버가 아니라 클래스에 고정된 멤버다. 그렇기에 클래스 로더가 클래스를 로딩해서 메소드 메모리 영역에 적재할 때 클래스별로 관리된다. 따라서 클래스의 로딩이 끝나는 즉시 바로 사용할 수 있다.

- **Constructor:** new 연산자와 같이 사용되어 클래스로부터 객체를 생성할 때 호출되어 객체의 초기화를 담당한다. 즉, ‘인스턴스 초기화 method’ 이다. 인스턴스 변수의 초기화 작업에 주로 사용되며, 인스턴스 생성 시에 실행되어야 할 작업을 위해서도 사용된다.

- **Overloading:** 한 가지의 class 내에 같은 이름의 method를 여러 개 정의하는 것으로 method overloading이라고도 한다.

- **Inheritance(상속):** Java에서 상속은 부모 클래스의 변수/메소드를 자식 클래스가 물려받아 그대로 사용 가능하게 해준다. 여기서 부모클래스를 superclass, 자식클래스를 subclass라 부른다. 상속은 **extends**라는 키워드를 사용하며 상속의 형태는 subclass extends superclass와 같다.

**- Interface:** 자바 프로그래밍 언어에서 클래스들이 구현해야 하는 동작을 지정하는데 사용되는 추상 자료형이다. 인터페이스는 마치 프로토콜과 비슷하다. 인터페이스는 interface라는 키워드를 사용하여 선언하며, 메서드 시그너처와 상수 선언(static과 final이 둘 다 선언되는 변수 선언)만을 포함할 수 있다.

- **Comparable Interface:** 정렬 수행 시 기본적으로 적용되는 정렬 기준이 되는 메서드를 정의하여 객체끼리 비교를 가능하게 해주는 인터페이스이다. 해당 인터페이스를 구현(implements)한 클래스는 반드시 compareTo() 메서드를 정의해야 한다.

- **Array.sort( ):** 지정된 개체 배열을 요소의 자연적 순서에 따라 오름차순으로 정렬한다. 배열의 모든 요소는 comparable interface를 구현해야 한다. 또한 배열의 모든 요소는 상호 비교 가능해야 한다(즉, e1.compareTo(e2)는 어레이의 e1 및 e2 요소에 대해 ClassCastException을 발생시키지 않아야 한다).

**- for() 반복문**: ()안의 조건이 만족되는 동안 반복문에 해당하는 코드를 반복 수행, 조건에 부합하면 break;

**Design:**

**#1**

1. Listing 8.2의 클래스 Student.java를 comparable interface로 수정해야한다.

🡪 **implements Comparable<Student>**

2. Array.sort( ) 메서드는 실행이 될 때 자동으로 comparable interface에 존재하는 compareTo( ) 메서드를 호출하게 된다. 따라서 Array.sort( )에 parameter로 들어가는 배열에 대해서 비교가 가능하도록 Student 클래스가 compareTo( )를 정의해야 한다.

🡪 **public int compareTo(Student s);**

3. 학생 번호를 기준으로 오름차순 정렬을 해야한다

🡪 **return** **this**.studentNumber - s.studentNumber; 를 통해 양수, 0, 음수의 integer value를 반환하게 된다.

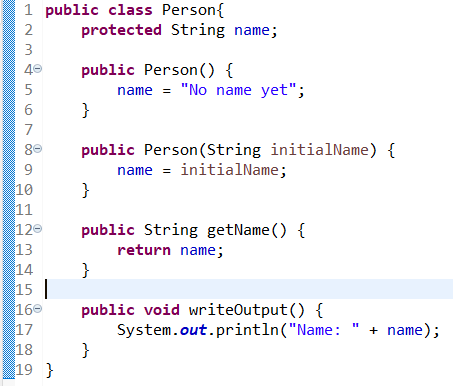
4. 학생 이름을 기준으로 사전식 오름차순으로 정렬을 해야한다.

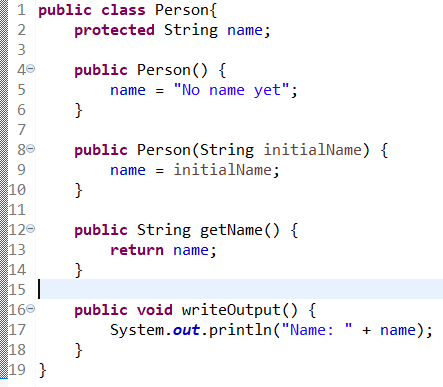
🡪 **return** name.compareTo(s.name);를 통해 양수, 0, 음수의 integer value를 반환하게 된다.

**Code:**

**#1**

**전체 코드**

**(StudentDemo.java)**

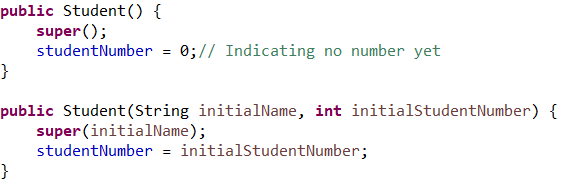
**(Person.java)**

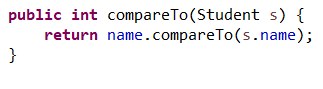
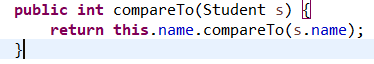
**(Student.java – 학생 번호를 기준으로 오름차순 정렬)**

**(Student.java – 학생이름을 기준으로 사전식 오름차순 정렬)**

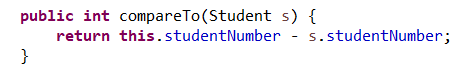
Person.java에 선언된 String 변수로 protected의 형태로 선언되었기 때문에 Person.java의 subclass는 이 변수를 사용이 가능하다.

Student.java는 public class로 다른 class에서 선언되어 사용하는 것이 가능하다. 또한 extends를 통해 Person.java에 선언된 class내부의 변수나 메서드 등을 사용하는 것이 가능하다. 그리고 implements를 통해 comparable interface로 선언되었다.

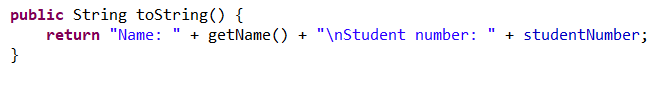
studentNumber는 private int로 선언되었기 때문에 Student 내부에서만 사용이 가능하다.

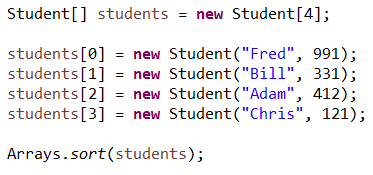
Student.java는 Person.java를 상속하고 있기 때문에 super( ) 메서드를 통해서 Person에 선언된 변수를 가져와 사용하는 것이 가능하다.

Student.java는 comparable interface이기 때문에 compareTo( ) 메서드를 필수적으로 정의하고 있어야 한다. 위의 코드는 학생의 이름을 기준으로 사전식 오름차순으로 정렬을 하도록 하는 compareTo( )메서드이다.

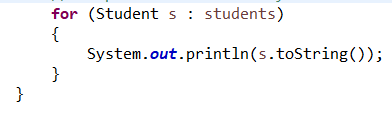
return 값으로 name.compareTo(s.name)을 반환하는데 이는 parameter로 받은 class의 String 변수 s.name의 값과 현재 class의 String 변수 name의 값을 비교하여 양수, 0, 음수의 값을 반환하게 되고 이것을 기준으로 오름차순 정렬이 가능하게 된다.

학생의 번호를 기준으로 오름차순으로 정렬하게 해주도록 정의가 된 compareTo( ) 메서드이다. return값으로 현재 class의 int 변수 this.studentNumber와 parameter로 받은 class의 int 변수 s.studentNumber의 차이를 구하여 양수, 0, 음수의 값을 반환하게 되고 마찬가지로 이 값을 기준으로 오름차순 정렬이 가능하게 된다.

Student.java에 정의된 메서드로 Person.java에서 상속된 getName( ) 메서드에서 return 받은 String 값과 현재 class의 int 변수인 studentNumber의 값을 출력하는 메서드이다.



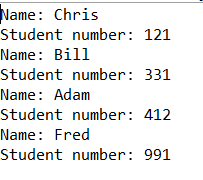
4개의 Student class를 저장하는 Class Array인 students를 선언하여 각 배열의 값을 선언하여 준다. 그리고 Arrays.sort(students) 메서드를 이용하여 오름차순 정렬을 시킨다. 이 때, Arrays.sort( ) 메서드는 students에 저장된 Students class의 compareTo( )메서드를 자동으로 호출하여 사용하고 compareTo( )에서 return 되는 값을 기준으로 정렬을 하게 된다.



for문의 조건식에서 students의 value를 가지는 Student s를 선언하고 s가 배열의 끝에 도달하기까지 toString을 통해서 그 값을 출력한다.

**Result:**

**#1 (학생의 번호를 기준으로 오름차순 정렬한 결과)**



**#2 (학생의 이름을 기준으로 사전식 오름차순 정렬한 결과)**

