

2018학년도 2학기
JAVA 프로그래밍 II

514770
2018년 가을학기
10/16/2018
박경신

과제 Lab5_1 (ArrayList & Generic & Inner Class)

1. Lab5_1은 Lab4를 ArrayList, generic, inner class, interface을 사용한 방식으로 수정한다.
2. interface Parser<T>
 1. T parse(String[] items); //
3. Parser를 상속받은 PersonParser를 구현한다.
 1. T parse(String[] items); // items 배열을 parse해서 Person 객체 생성
4. Parser를 상속받은 AddressBookParser를 구현한다.
 1. T parse(String[] items); // items 배열을 parse해서 AddressBook 객체 생성
5. Lab5_1에서는 Manager 클래스의 메소드를 테스트하는 프로그램을 작성한다.

과제 Lab5_1 (ArrayList & Generic & Inner Class)

5. class Manager<T> implements Iterable<T>를 구현한다.
 1. private List<T> data = null;
 2. private Parser parser = null;
 3. private class ManagerIterator<T> implements Iterator<T> { /* 내부 클래스 구현 */ }
 4. public Manager(Parser parser); // 생성자
 5. public List<T> getAll(); // 모든 데이터 리스트로 반환
 6. public T get(int index); // index에 있는 데이터 반환
 7. public void add(T d); // 데이터를 마지막 위치에 추가
 8. public void insert(int index, T d); // 데이터를 index 위치에 삽입
 9. public void addAll(List<T> data); // 새 데이터를 기존 리스트에 더 추가
 10. public void remove(int index); // 배열 해당 인덱스 객체를 지우고, 이후 데이터를 앞당김
 11. public void removeAll(); // 배열에 모든 객체 삭제
 12. public Iterator<T> iterator(); // iterator 객체를 반환
 13. public void saveCSV(String filename); // 데이터를 파일로 저장
 14. public void loadCSV(String filename); // 내부적으로 parser를 이용하여 데이터를 로딩

과제 Lab5_2 (dynamic array)

1. class Manager<T> implements Iterable<T>의 내부 구현을 Dynamic Array (ArrayList 같은) 방식으로 구현한다.
 1. private T[] data = null;
 2. private int count = 0;
 3. private int size = 0;
 4. private Parser parser = null;
 5. private class ManagerIterator<T> implements Iterator<T> { /* 내부 구현 */ } // count 사용
2. Lab5_2에서는 Manager 클래스의 메소드를 테스트하는 프로그램을 작성한다. 그리고, ArrayList와 Dynamic Array 방식을 비교한다.

과제 Lab5_2 (dynamic array)

- class Manager<T> implements Iterable<T>를 구현한다.
 - private void copy(T data, int newSize); // 배열복사 후 마지막에 새데이터 추가
 - private void copy(int index, T data, int newSize); // 중간에 새데이터 추가 배열복사
 - public Manager(int size, Parser parser); // 생성자**
 - public T[] getAll(); // 모든 데이터 배열로 반환
 - public T get(int index); // index에 있는 데이터 반환
 - public void add(T data); // copy(T,int) if array size is dynamically increased
 - public void insert(int index, T data); // copy(int, T,int) if array size is dynamically increased
 - public void addAll(T[] data); // 새 배열을 기존 배열에 더 추가
 - public void remove(int index); // 배열 해당 인덱스 객체를 지우고, 이후 데이터를 앞당김
 - public void removeAll(); // 배열에 모든 객체 삭제
 - public Iterator<T> iterator(); // iterator 객체를 반환**
 - public void saveCSV(String filename); // 배열을 파일로 저장
 - public void loadCSV(String filename); // 내부적으로 parser를 이용하여 데이터를 로딩**

과제 Lab5_3 (==, equals, hashCode, Collection)

- Person, YouTuber, Foreigner 클래스에 equals과 hashCode를 override하라.
- AddressBook, 등등 클래스에 equals과 hashCode를 override하라.
- Manager<T> 클래스에서는 다음 메소드를 추가한다.
 - public boolean contains(T data);
 - public void sort(); // sort by Comparable
 - public void sort(Comparator<? super T> comparator); // sort by Comparator
- Lab5_3 클래스에서는 ==, equals, hashCode를 테스트하는 프로그램을 작성한다.
- 그리고, Lab5_3 클래스에서는 Vector, ArrayList, LinkedList, HashSet, HashMap를 사용하여 테스트하고 자료구조를 비교 분석하라.

과제 Lab5_4 (lambda, stream, query)

- Manager<T>에 다음 메소드를 추가한다. 구현은 lambda, stream을 사용한다.
 - public List<T> findIf(Predicate<? super T> predicate); // return found elements into another list
 - public List<String> findIf(Predicate<? super T> predicate, Function<? super T, String> func); // return found elements into a String list
 - 추가적으로 본인이 3개 이상의 query 메소드를 추가한다.
- Lab5_4에서는 Manager 클래스의 findIf, findNameIf 등등 메소드를 테스트 하는 프로그램을 작성한다. Lambda와 stream을 이용한다.
- Lab5_4에서 5개 이상의 본인만의 query테스트를 작성하라. Lambda와 stream을 이용한다.

과제 제출

- Lab5_1 ~ Lab5_4와 보고서를 전체적으로 묶어서 e-learning에 과제 제출