

Java Programming II

Lab7

514770-1

Fall 2023

11/7/2023

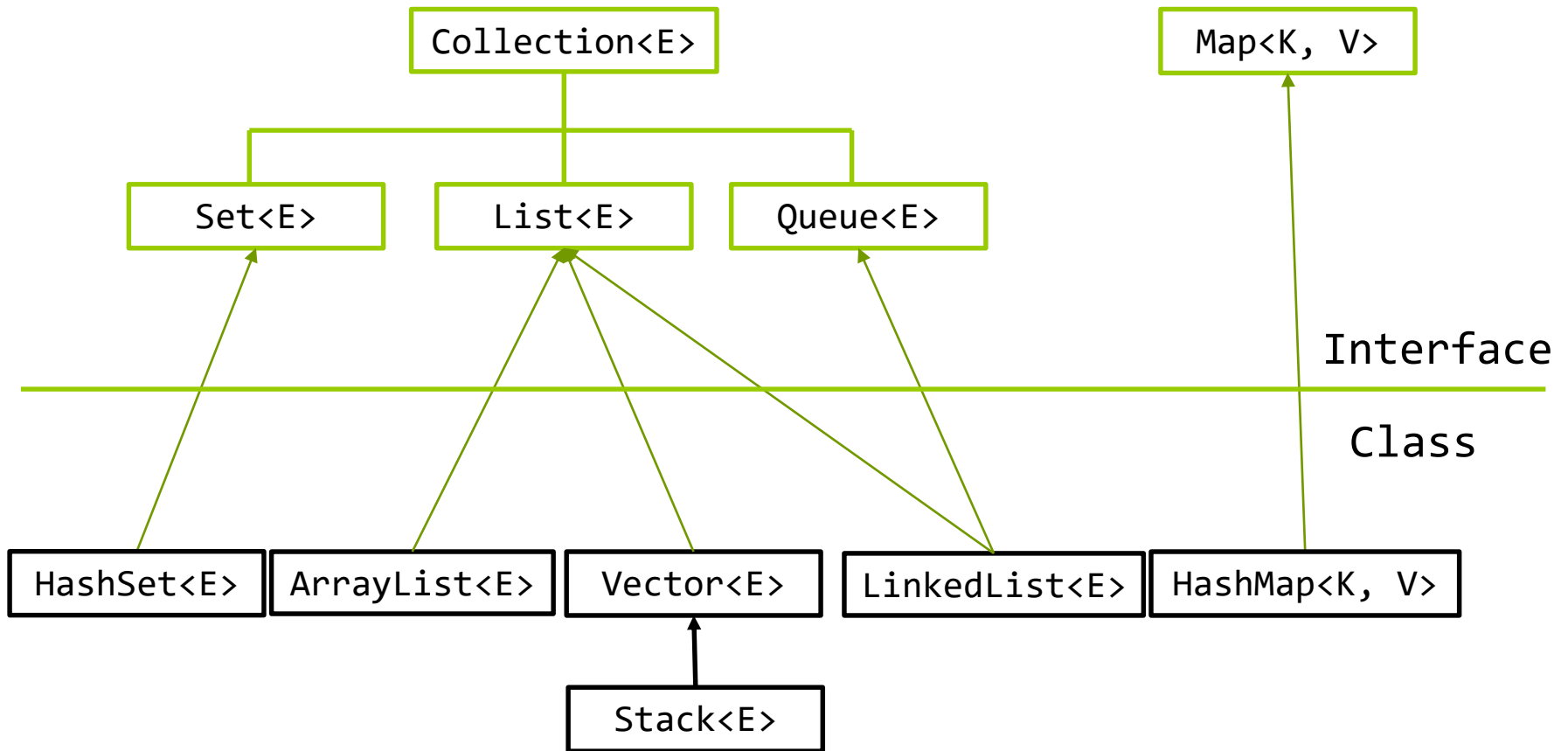
Kyoung Shin Park
Computer Engineering
Dankook University

Lab7

- Practice to write a program that adapts `List<E>` to `DataCollection<E>` and `FileLoader<E>` to `FileImporter<E>`.
 - `java.util.List<E>` interface
 - `ArrayList`, `LinkedList`, `Vector`, `Stack` class implements `List<E>`.
 - `DataCollection<E>` interface extends `Iterable<E>`.
 - `DynamicArray<E>` class implements `DataCollection<E>`.
 - `ListDataCollection<E>` adapter class takes `List<E>` adaptee class to support `DataCollection<E>` target interface.
 - `FileImporter<E>` interface has `List<E> importFile(String filepath)` and `void exportFile(String filepath, List<E> list)`.
 - `ElementCSVImporter` class implements `FileImpoter<Element>`.
 - `FileLoader<E>` interface has `List<E> load(String filepath)`.
 - `ElementJSONLoader` class implements `FileLoader<Element>`.
 - `ElementXMLLoader` class implements `FileLoader<Element>`.
 - `FileLoaderImporterAdapter<E>` adapter class takes `FileLoader<E>` adaptee class to support `FileImporter<E>` target

Lab7

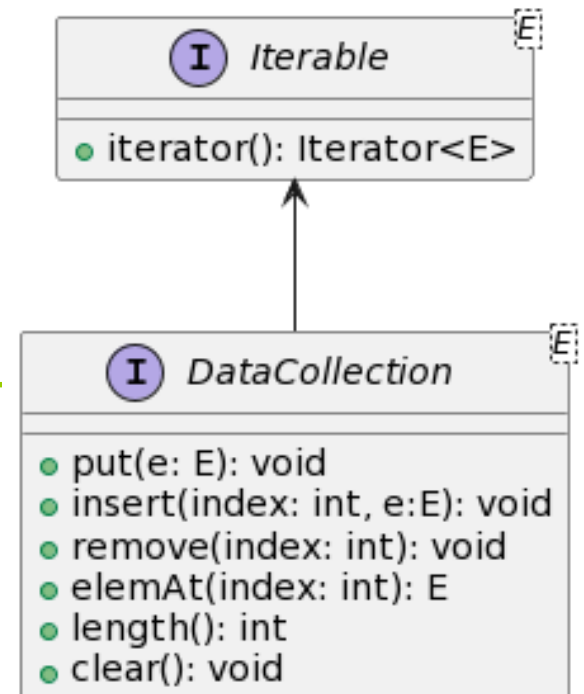
□ java.util.Collection



Lab7

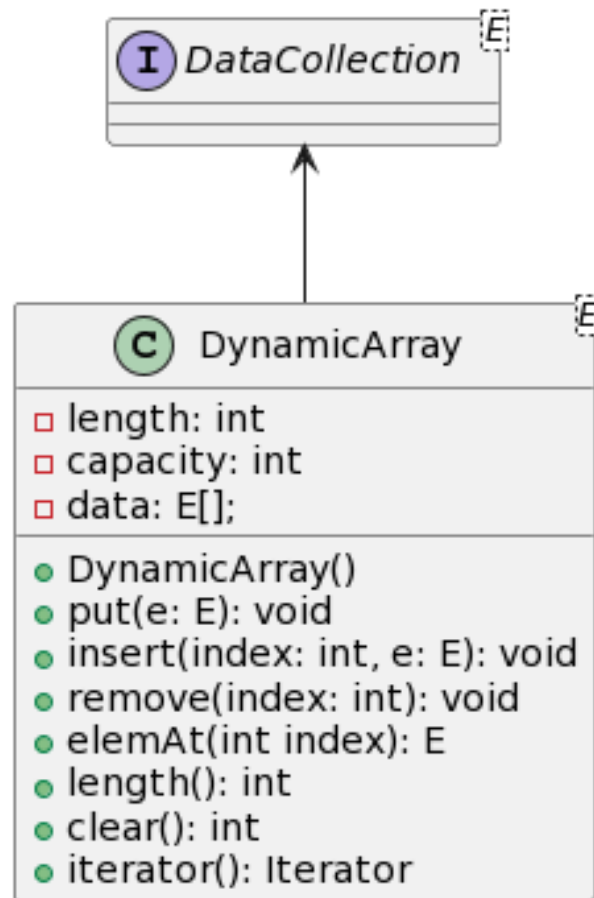
- **DataCollection<E> interface extends Iterable<E> - foreach**

```
public interface DataCollection<E> extends Iterable<E> {  
    void put(E e);  
    void insert(int index, E e);  
    void remove(int index);  
    E elemAt(int index);  
    int length();  
    void clear();  
}
```



Lab7

- **DynamicArray<E>** class implements **DataCollection<E>**
 - A generic dynamic array class using E[] data



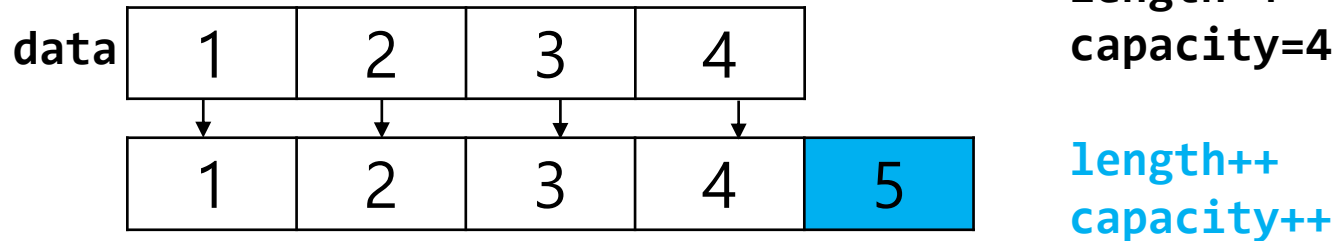
Lab7

- **DynamicArray<E>** class implements **DataCollection<E>**.
 - `public void put(E e); // put element to the dynamic array`
 - `public void insert(int index, E e); // insert element at the index`
 - `public void remove(int index); // remove element at the index`
 - `public E elemAt(int index); // get element at the index`
 - `public int length(); // get the length of dynamic array (# of elements)`
 - `public void clear(); // remove all elements and reset`
 - **`public Iterator<E> iterator(); // returns
DynamicArrayIterator<E>()`**
- **DynamicArrayIterator<E>** class implements **java.util.Iterator<E>**.
 - `public boolean hasNext();`
 - `public E next();`
 - `public void remove();`

Lab7

□ Dynamic Array

- Put an element at the end of an array



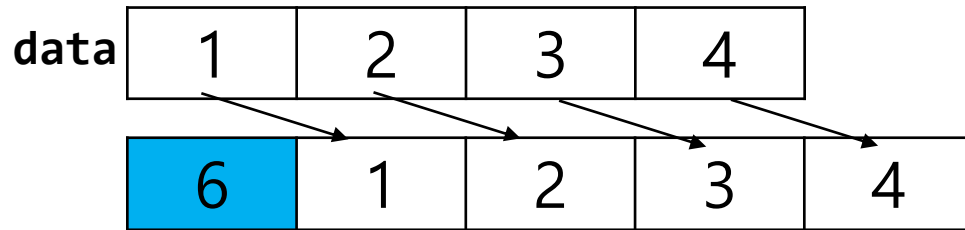
- Put an element at the end of an array



Lab7

□ Dynamic Array

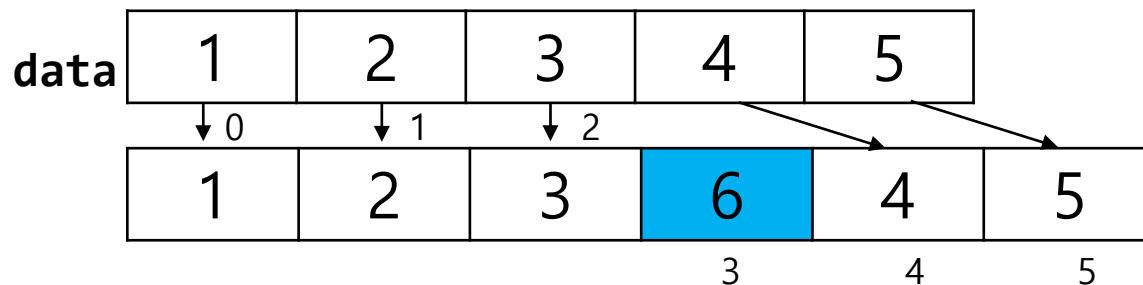
- Insert an element at the index 0



length=4
capacity=4

length++
capacity++

- Insert an element at the index 3



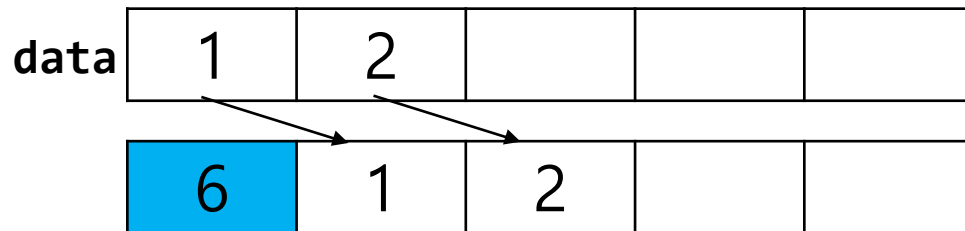
length=5
capacity=5

length++
capacity++

Lab7

□ Dynamic Array

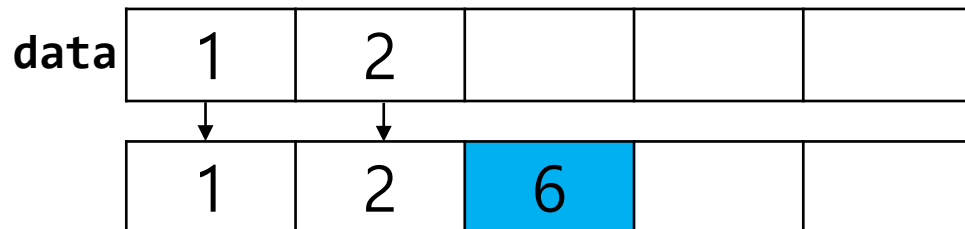
- Insert an element at the index 0



length=2
capacity=5

length++
capacity=5

- Insert an element at the index 2 – If (index > length) `ArrayIndexOutOfBoundsException`



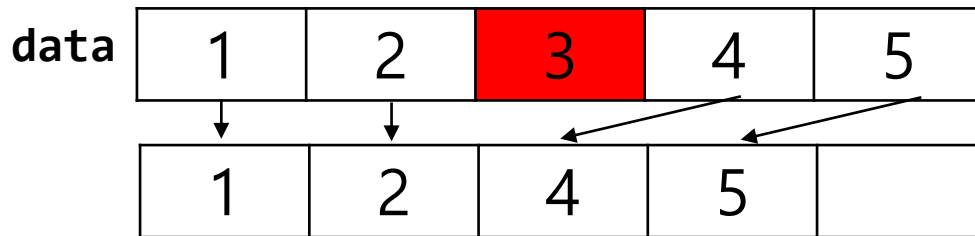
length=2
capacity=5

length++

Lab7

□ Dynamic Array

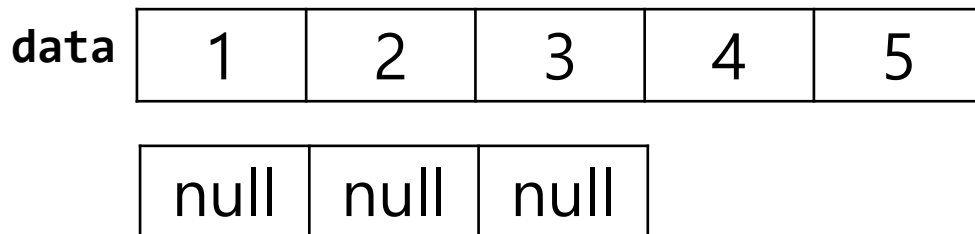
- Remove an element at the index 2



length=5
capacity=5

length--

- Clear

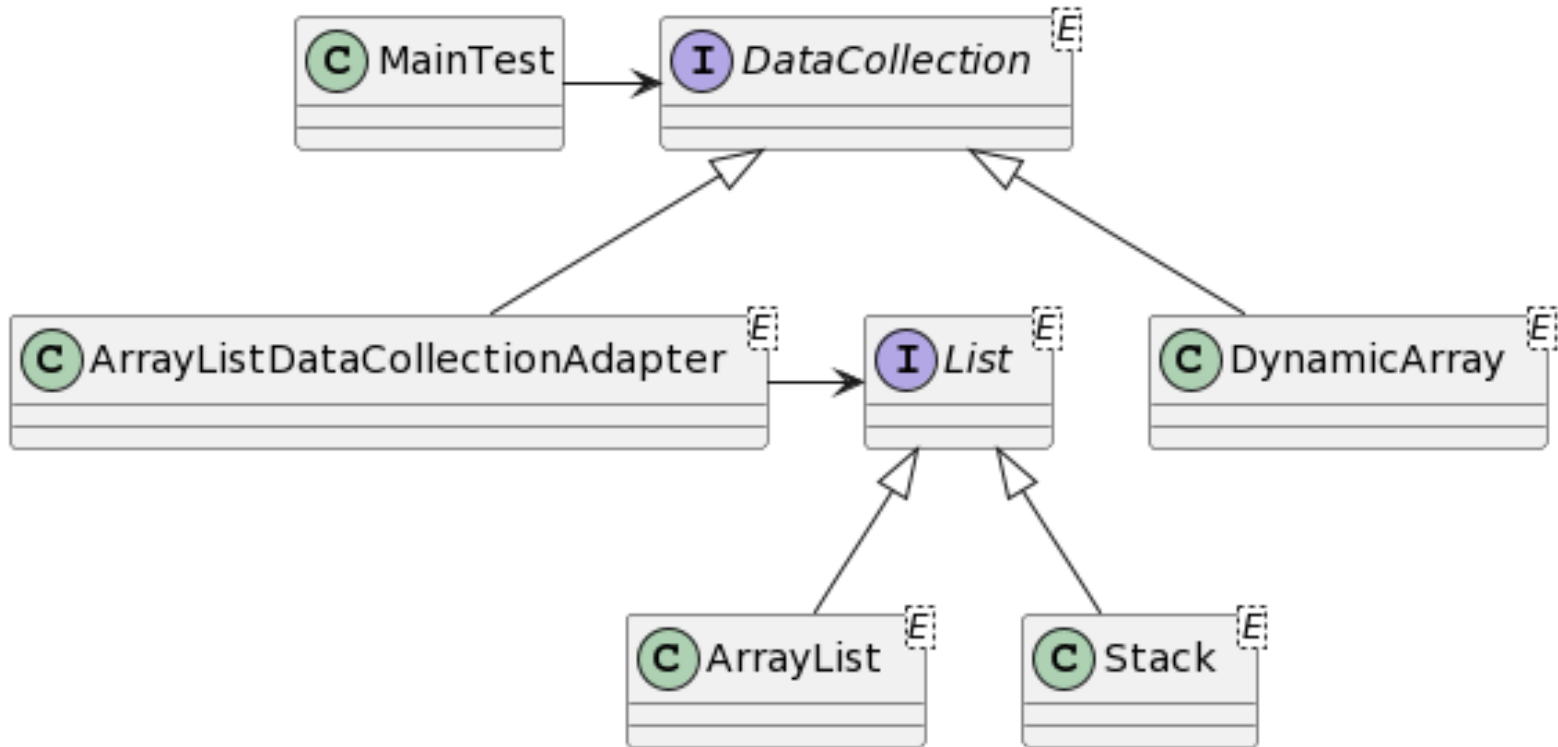


length=5
capacity=5

length=0 // reset
capacity=SIZE // reset

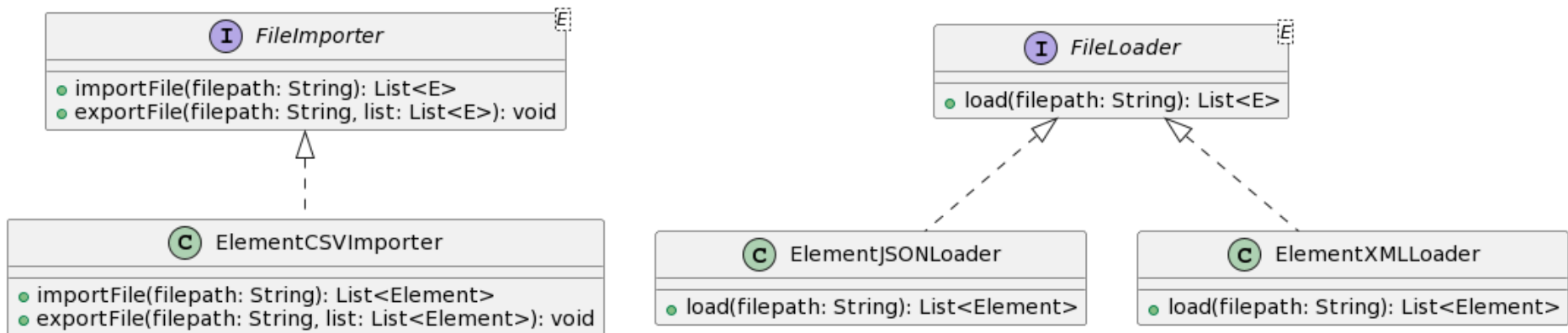
Lab7

- **ListDataCollection adapter** class takes the **List adaptee** class to support the **DataCollection target** interface.



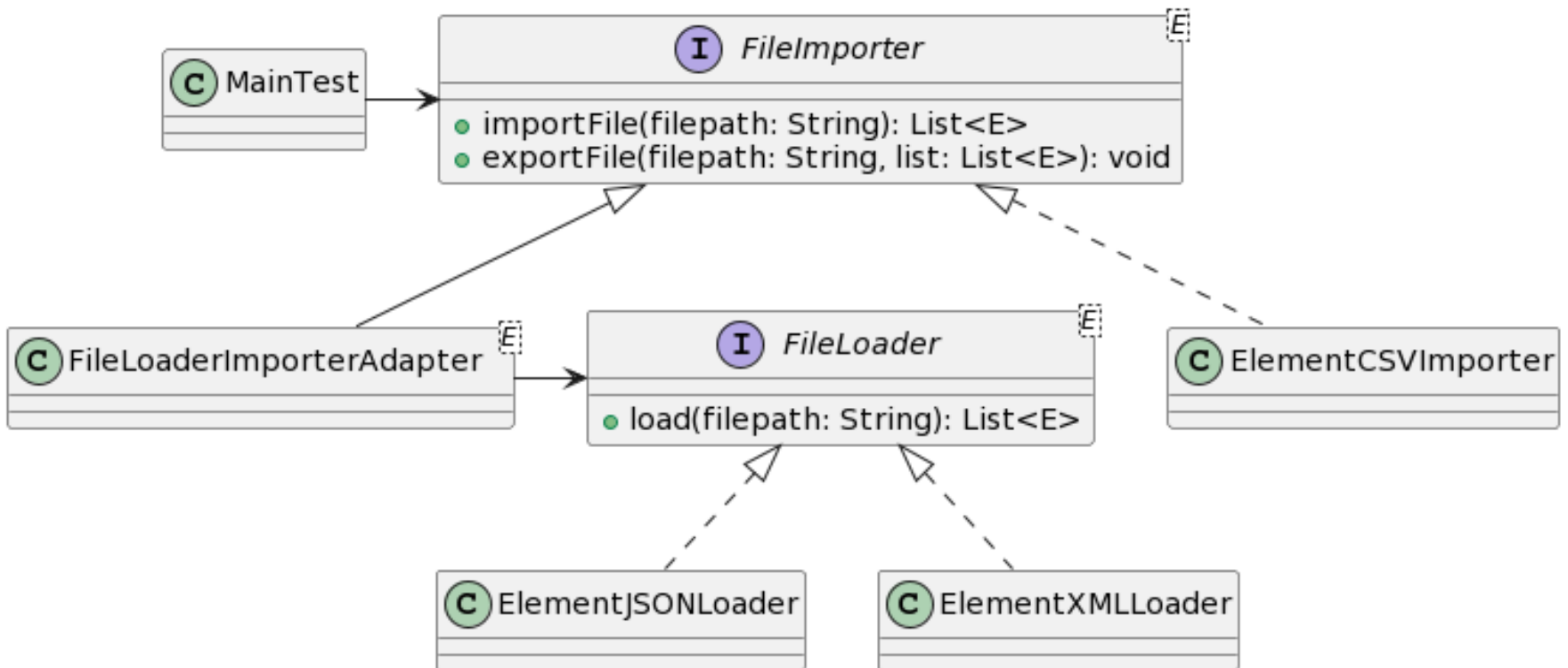
Lab7

- **FileImporter<E>** interface
 - **ElementCSVImporter** class implements FileImporter<Element>
- **FileLoader<E>** interface
 - **ElementJSONLoader (using com.google.gson.Gson)** class implements FileLoader<Element>
 - **ElementXMLLoader (using org.w3c.dom.Element/Document/NodeList)** class implements FileLoader<Element>



Lab7

- FileLoaderImporterAdapter<E> adapter class takes FileLoader<E> adaptee class to support FileImporter<E> target interface.



Lab7

```
public class MainTest {
    public static void main(String[] args) {
        List<PeriodicElement> list =
PeriodicElementImporter.loadCSV("PeriodicElements.csv");
        System.out.println("\n\nDynamicArray add & print");
        DataCollection<Element> arr = new DynamicArray<>();
        for (PeriodicElement e : list) {
            arr.put(new Element(e.getNumber(), e.getName(), e.getSymbol(),
e.getWeight()));
        }
        arr.forEach(System.out::println); // test Iterable<E>
        // remove & insert & elemAt & clear & remove all using iterator &
print using for/while/foreach
        List<Element> elements = new ArrayList<>();
        list.forEach(e -> elements.add(new Element(e.getNumber(),
e.getName(), e.getSymbol(), e.getWeight())));
        elements.forEach(System.out::println);
    }
}
```

Lab7

```
DataCollection<Element> arr2 =  
    new ListDataCollectionApdapter<>(elements);  
    // remove & insert & elemAt & clear & remove all using iterator &  
    print using for/while/foreach  
  
Stack<Element> stack = new Stack<>();  
// stack push all elements  
DataCollection<Element> arr3 =  
    new ListDataCollectionApater<>(stack);  
// remove & insert & elemAt & clear & remove all using iterator &  
    print using for/while/foreach  
// ...
```

Lab7

```
// ElementJSONLoader
FileImporter<Element> importer =
    new FileLoaderImporterAdapter<>(new ElementJSONLoader());
List<Element> elements2 = importer.importFile("Elements.json");
elements2.forEach(System.out::println);
// ElementXMLLoader
importer =
    new FileLoaderImporterAdapter<>(new ElementXMLLoader());
elements2 = importer.importFile("Elements.xml");
elements2.forEach(System.out::println);
// ElementCSVImporter
importer = new ElementCSVImporter();
elements2 = importer.importFile("Elements.csv");
elements2.forEach(System.out::println);
importer.exportFile("Elements1.csv", elements2);
}
}
```


Submit to e-learning

- Add your code (e.g., test all methods with DataCollection using ArrayList, Stack) in the Lab7 assignment.
- Submit the Lab7 assignment (JAVA23-2-Lab7-YourID-YourName.zip including the report) to e-learning.