Tut-Lab Week 5

Aims:

- Understand polymorphism and interface type design
- Gain practical experience implementing an interface type
- Learn to evaluate different class implementations
- Use and understand different notions of equality in Java

Preparation:

- Review the definitions of sets and their basic operations

Sets

- Define a **Set<E>** interface type that can handle elements of a generic type **E**
    - Consider operations on set membership (adding and removing elements, testing membership), and basic operations on sets (union, intersection, subset)
    - Include an equals method for sets
- Define a class realizing the **Set<E>** interface that uses an **ArrayList<E>** to store the elements
    - **Pay particular attention to equality of sets and set elements**
- Explain how your code enforces the class invariant that all elements of a set are distinct
- By having **Set<E>** implement **Iterable<E>**, define a method iterator() than returns an iterator over sets
    - This method is straightforward (one line of Java)
- Write a test class for **Set<E>** that uses a **Scanner** to read elements from an input file, then add them to various sets (e.g. of type **String)**
    - Each line of the input should contain the elements of one set