

CSC301

Iterators & Lambda Expressions

Iterators

- An iterator is an object that lets you traverse (i.e. go through) items in a collection, without accessing the collection itself.
 - Extremely common *design pattern*
 - Interface might vary slightly between programming languages

Iterator in Java

- Generic interface Iterator<T>
 - `hasNext()`
 - Returns a boolean
 - Indicates whether there is a next item
 - `next()`
 - Return the next item (of type `T`)
 - The type of the item depends on the collection we're iterating over.
In Java, you can define a generic collection.

Iterables

- We usually distinguish between two concepts:
 - Iterable<T>, a collection of items that can be traversed using an iterator.
 - Iterator<T>, a “utility object” used for traversing an iterable collection.
- In Java, you can use iterables in a for-each loop

```
for (T item : iterable) {  
    // Loop body ...  
}
```

- *Note:* Iterables are **not** a Java-specific concept. For example, the same distinction exists in Python (although it is sometimes a little less clear).

Why Iterators?

- Modularity
 - Changing an underlying collection (e.g. instead of a list, use a tree or a set) does not require changes in other pieces of the code.
- In some cases, memory efficiency
 - Generate a large (or even unbounded) sequence of items, using little memory space.
- Convenience
 - Abstract implementation details such as network communication, caching or lazy-evaluation.
Ex: Infinite scrolling, database cursor
- Clear & Explicit Design
 - Indicates that your code only needs a way to traverse the items, but doesn't need

Code Examples

- [Range of integers](#)
- [Custom iterator](#)

Lambda Expressions

- Lambda expressions were introduced to Java in version 8
 - Conveniently define a function inline
 - Avoid the need to create (anonymous) class, when all you need is a function
 - Together with Functional Interfaces, they add *functional programming* capabilities to Java
- The same concept exists in many programming languages

Code Examples

- Filtering Iterator that uses a Predicate
- Mapping iterator that uses a Function
- Use lambda expressions together with the filtering and mapping iterators