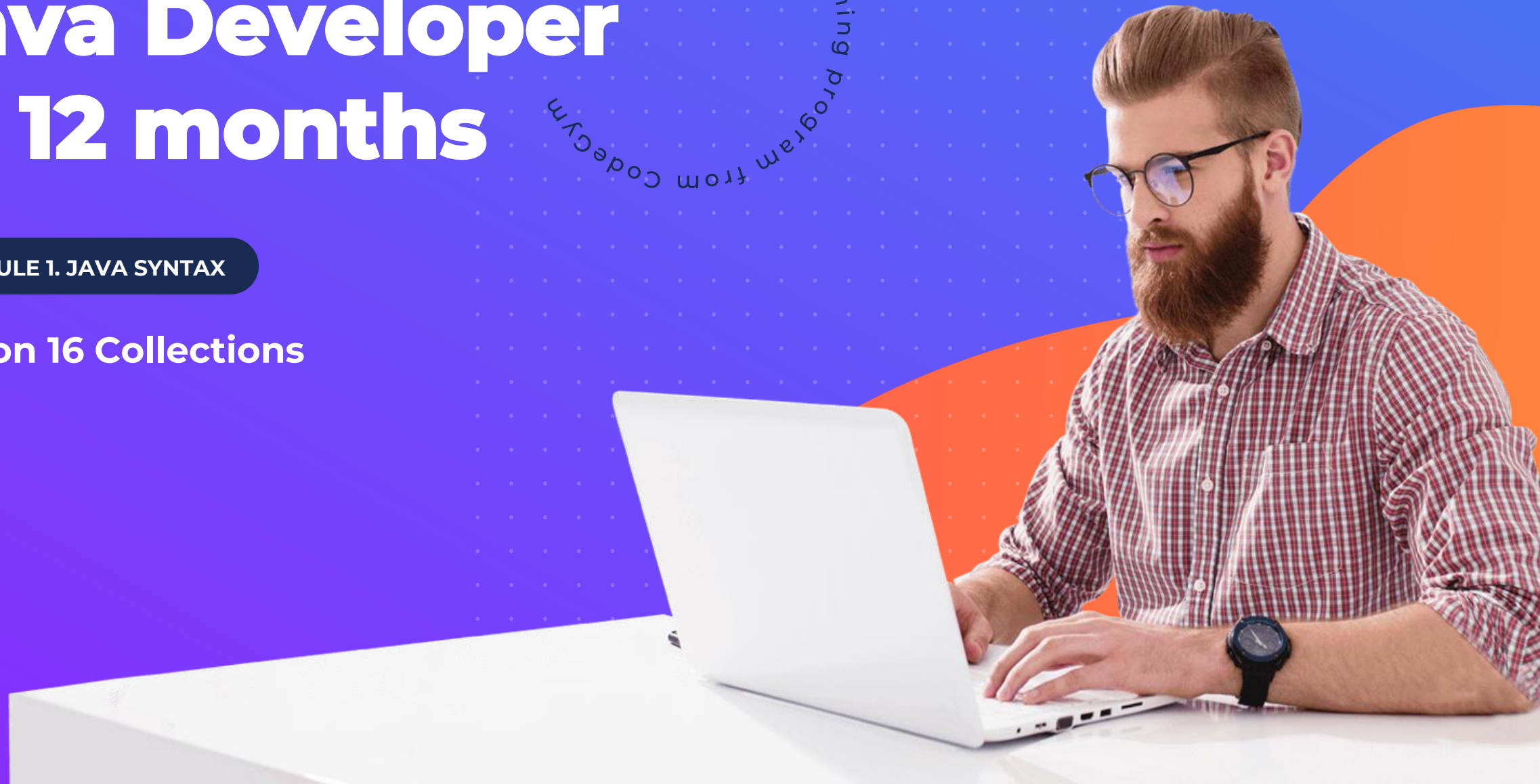
CODEGYM

Java Developer in 12 months

Mentor-supported training program from CodeGym

MODULE 1. JAVA SYNTAX

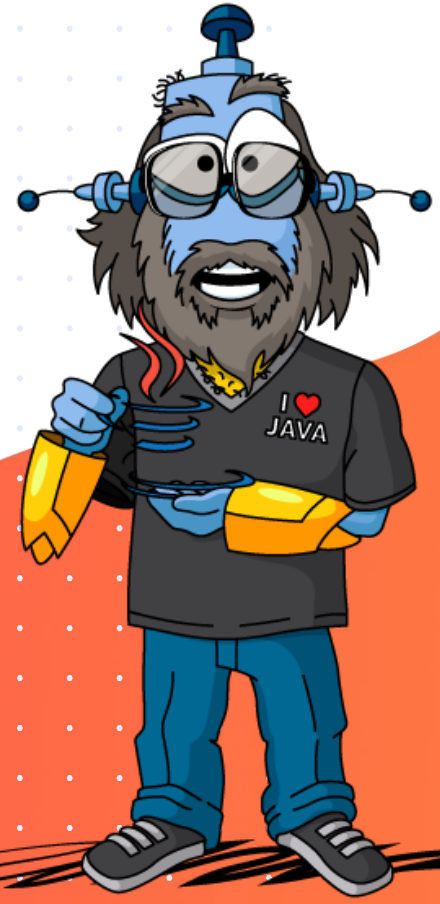
Lesson 16 Collections



CODEGYM

Lesson plan

- Containers and collections
- Set
- HashSet Collection
- Iterator
- For-each loop
- Removing an element in a for-each loop



CODEGYM

Containers and collections

Containers or collections are classes that let you store and process several objects at once.

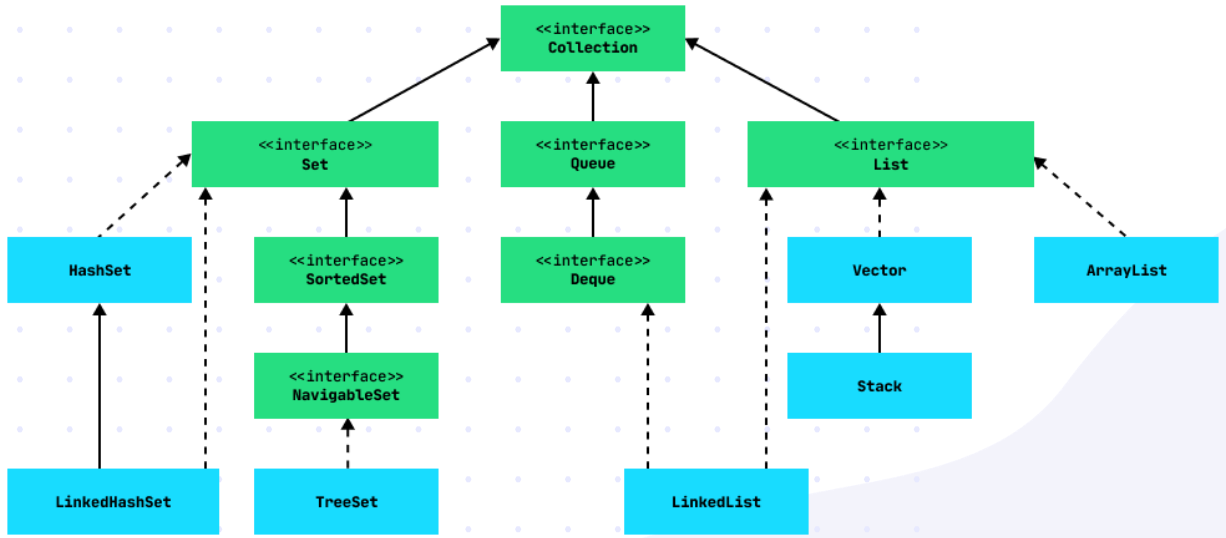
You already know two kinds of containers: arrays and lists.

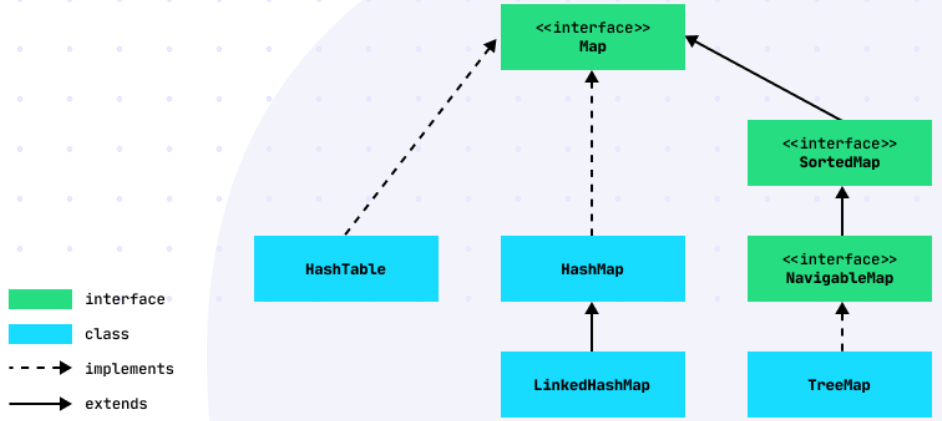
Java has many collections, each of which stores elements in its own specific way.

Collection	Class	Description
List	ArrayList	List
	LinkedList	Linked list
	Vector	Vector
	Stack	Stack
Set	HashSet	Set
	TreeSet	
	LinkedHashSet	
Queue	PriorityQueue	Queue
	ArrayQueue	
Map	HashMap	Map/Dictionary
	TreeMap	
	HashTable	

CODEGYM

Collection hierarchy





CODEGYM

Set

The Set collection is designed to hold a set of elements. That's why it's called a Set (set). This collection has three features.

Operations on a set

There are only three operations:

- add elements to the set
- remove elements from the set
- check whether the set contains a specific element

No order

Elements in this collection do not have indices.

Unique elements

All elements in a set are unique.

CODEGYM

HashSet collection

The HashSet class is a typical set collection. You can create a HashSet object using a statement like:

```
HashSet<Type> name = new HashSet<>();
```

The HashSet class has methods like this:

Method	Description
boolean add(Type value)	Adds the value element to the collection
boolean remove(Type value)	Removes the value element from the collection. Returns true if there was such an element
boolean contains(Type value)	Checks whether the collection has a value element
void clear()	Clears the collection, removing all the elements
int size()	Returns the number of elements in the collection

CODEGYM

Iterator

An iterator is a special object associated with a collection, which helps traverse all the elements of the collection without repeating any.

You can use the following code to get an iterator for any collection:

```
Iterator<Type> it = name.iterator();
```

Where name is the name of collection variable, Type is the type of the elements of the collection, iterator() is one of the collection's methods, and it is the name of the iterator variable.

An iterator object has 3 primary methods:

Method	Description
Type next()	Returns the next element in the collection
boolean hasNext()	Checks whether there are any elements that have not been traversed yet
void remove()	Removes the current element of the collection

CODEGYM

For-each loop

It is used to traverse all elements of a collection.
This is how it looks in general:

```
for(Type name:collection)
```

Where collection is the name of the collection variable, Type is the type of the elements in the collection, and name is the name of a variable that takes the next value from the collection at each iteration of the loop.

This kind of loop iterates through all the elements of a collection using an implicit iterator.

CODEGYM

Remove an element in a for-each loop

The for-each loop has one drawback: **it can't remove elements correctly.**

There are three ways to get around this limitation:

1. Use a different kind of loop
2. Use an explicit iterator
3. Use a copy of the collection

Important!
You can't change a collection while you are traversing it with an iterator.

Example:
Use a copy of the collection

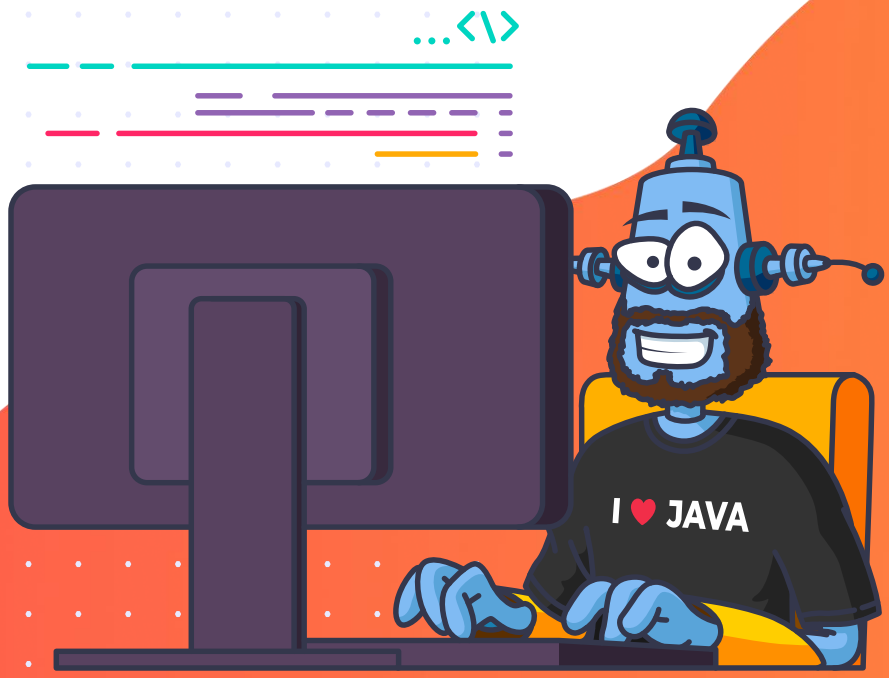
```
public static void main(String[] args) {
    //TODO: remove negative elements
    Set<Integer> numbers = new HashSet<>();
    numbers.add(7);
    numbers.add(-2);
    numbers.add(-7);
    numbers.add(14);
    numbers.add(4);
    System.out.println(numbers);

    Set<Integer> copyNumbers = new HashSet<>(numbers);
    for (Integer number : copyNumbers) {
        if (number < 0) {
            numbers.remove(number);
        }
    }
    System.out.println(numbers);
}
```

CODEGYM

Homework

MODULE 1. JAVA SYNTAX

Complete Level 17



CODEGYM

Answers to questions

MODULE 1. JAVA SYNTAX