```
❯ jshell ImList.java
|  Welcome to JShell -- Version 17.0.2
|  For an introduction type: /help intro

jshell> ImList list = new ImList()
list ==> []

jshell> list.add(1)     using add on list that is not generic throws an error
|  Warning:
|  unchecked call to add(T) as a member of the raw type ImList
|  list.add(1)
|  ^---------^
$7 ==> [1]

jshell> list     immutability
list ==> []

jshell> list = list.add(1)
|  Warning:
|  unchecked call to add(T) as a member of the raw type ImList
|  list = list.add(1)
|         ^---------^
list ==> [1]

jshell> int num = list.get(0)
|  Error:
|  incompatible types: java.lang.Object cannot be converted to int
|  int num = list.get(0);
|            ^---------^

jshell> int num = (int) list.get(0)
num ==> 1

jshell> ImList<Integer> list2 = list
|  Warning:
|  unchecked conversion
|    required: ImList<java.lang.Integer>
|    found:    ImList
|  ImList<Integer> list2 = list;
|                          ^--^
list2 ==> [1]

jshell> list2.set(0,99)
$12 ==> [99]

jshell> list
list ==> [1]

jshell> list2 = list2.set(0, 99).add(2)    using add on generic list2 no longer throw error
list2 ==> [99, 2]

jshell> ImList<Integer> list3 = new ImList<>().addAll(list2)
|  Error:
|  incompatible types: ImList<java.lang.Object> cannot be converted to ImList<java.lang.Integer>
|  ImList<Integer> list3 = new ImList<>().addAll(list2);
|                          ^--------------------------^

jshell> new ImList<>().addAll(list2)  the ImList here is actually a raw type, i.e. not generic!
$15 ==> [99, 2]

jshell> ImList<Integer> list3 = new ImList<Integer>().addAll(list2)    addAll
list3 ==> [99, 2]

jshell> class AscendingComp implements Comparator<Integer> {      writing your own Comparator
   ...>     @Override
   ...>     public int compare(Integer a, Integer b) {
   ...>         return a - b;
   ...>     }
   ...> }
|  created class AscendingComp

jshell> class DescendingComp implements Comparator<Integer> {
   ...>     @Override
   ...>     public int compare(Integer a, Integer b) {
   ...>         return b - a;
   ...>     }
   ...> }
|  created class DescendingComp

jshell> ImList<Integer> list4 = new ImList<>(List.of(3,6,8,0,99,-5,-1000))    second constructor
list4 ==> [3, 6, 8, 0, 99, -5, -1000]

jshell> list4.sort(new AscendingComp())
$20 ==> [-1000, -5, 0, 3, 6, 8, 99]

jshell> list4.sort(new DescendingComp())
$21 ==> [99, 8, 6, 3, 0, -5, -1000]

jshell> list4.sort(Comparator.reverseOrder())   built-in method Comparator.reverseOrder()
$22 ==> [99, 8, 6, 3, 0, -5, -1000]

jshell> for (int num: list4) {
   ...>     System.out.println(num * num);    Enhanced for loop, no need to use .get(index)
   ...> }
9
36
64
0
9801
25
1000000
```