Generics & collection set & list
Wrapper class

→ IS java pure OOps?
Ans:- because of the primitive data types java is not pure oops it is fully oops

pure oops:
java script
python

int, float double, long

for every data type we have pre defined class
int is Integer

new :- allocates the memory.

```
int a = 10;
Integer i = new Integer(a); // boxing
```

wrapper

```
Integer x = 5; // boxing
Integer k = new Integer(100);
int b = k; // unboxing
```

→ when we use wrapper class, if directly object into data type it convert into autoboxing

generics :- unknown type single word

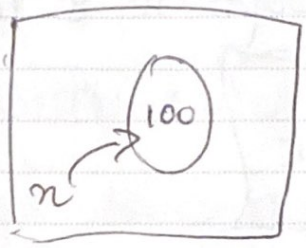
→ In generic, writing without any type is unsafe.

```
Example x = new Example(); // unsafe
x.show(30);
x.show(4.3);
```

// gaye

```
Example <Integer> i = new Example <Integer>();
  i.show(100);
Example <Double> b = new Example <Double>();
  b.show(4.5);
```

T x;
x is a reference
1. show (100)
1. show (new Integer (100))



~~Part~~

Array:
→ collection of similar data types.

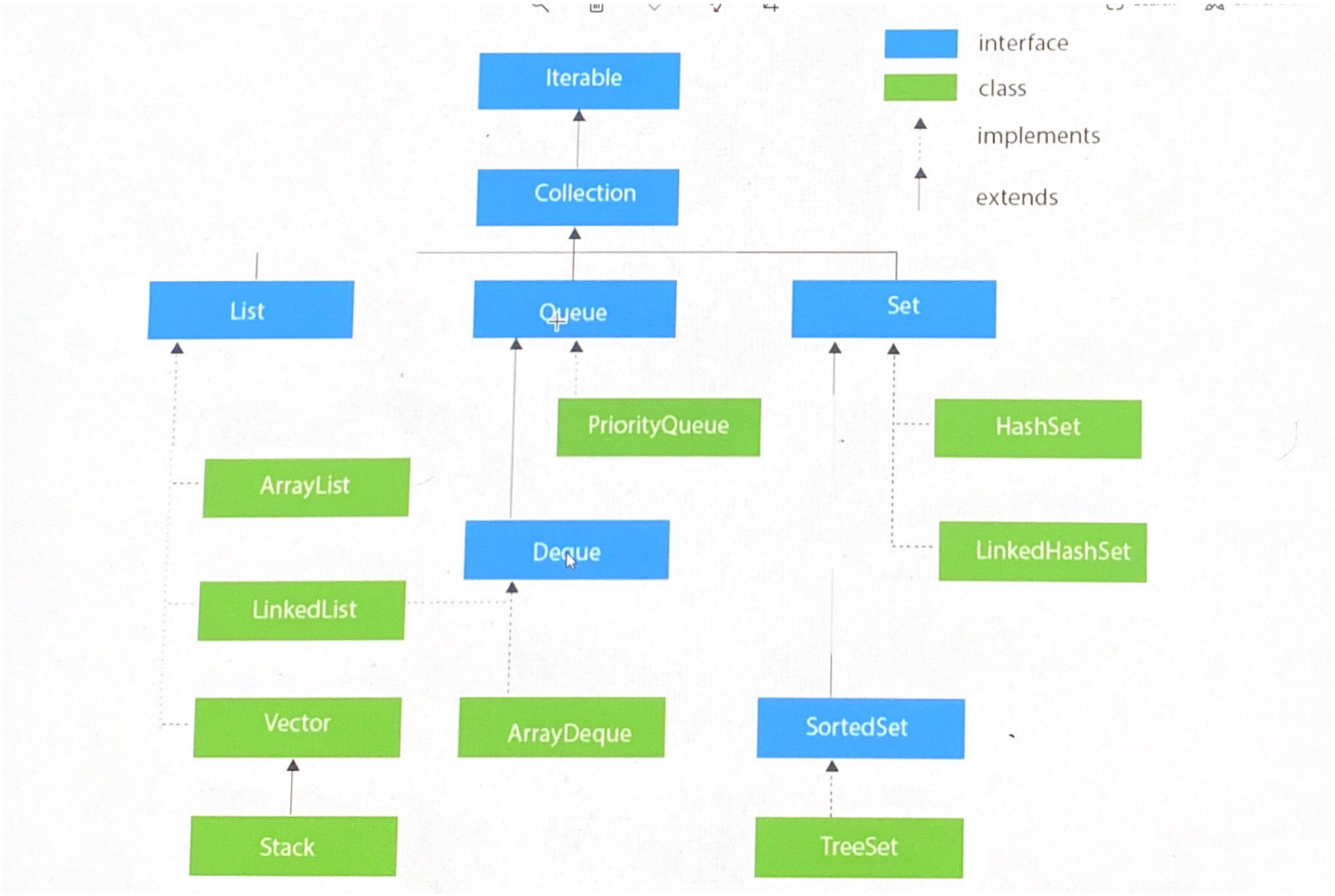
drawbacks of Array
1. fixed size
2. can't store disimilar data.

collection: is a interface
→ a group of individual objects.

note: with the help of generics and wrapper classes

Set:
→ It is also a group of individual objects

→ It doesn't allow duplicate values.
→ It allows null values.
→ It allows heterogenous elements.


interface
class
implements
extends
Iterable
Collection
List
Queue
Set
PriorityQueue
HashSet
ArrayList
Deque
LinkedHashSet
LinkedList
Vector
ArrayDeque
SortedSet
Stack
TreeSet

HashSet: It follows random order

LinkedHashSet: It follows insertion order

TreeSet :- It follows ascending order
→ It doesn't allow heterogenous elements

System.out.println(hs.size());
→ size gives the sizes but it doesn't count the duplicate.

boolean isEmpty();
boolean contains(java.lang.Object);
boolean add(E);
boolean remove(java.lang.Object);
void clear();

TreeSet has ~~a subset~~
:-) subSet[E, boolean, E, boolean);

→ headSet [E, boolean);
→ tailSet [E, boolean);

There is NavigableSet & SortedSet of the subSet, headSet & tailSet

Note:- for headSet boolean is false by default
→ for tailSet, Boolean is true by default.

example
System.out.println (hs.subSet ("M", false, "S", true));
(if you don't want to include)
(if you want to include)

list
→ It is also a group of individual objects.
→ It allows duplicate values.
→ It allows null values
→ It allows heterogenous elements.

Arraylist
→ follow insertion order
→ duplicate is allowed
→ select * from employee; // better for retrieving all records

linkedlist
→ follow insertion order
→ duplicate is allowed
→ select * from employee where id = 1023;
// for searching linkedList is better

Vector
→ follows insertion order
→ allows duplication
→ All the methods are synchronized

→ In list we have stack also
→ Stack have
↳ push(E);
pop();
peep();
emty();

all the methods of vector are available in stack