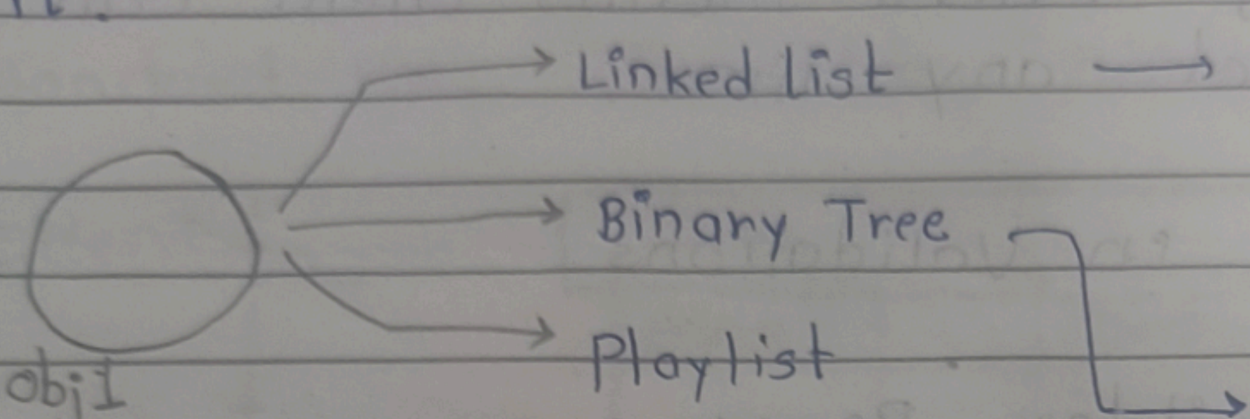
Lecture 29 : Iterator Design Pattern

Page No.
Date

Introduction

- It is used by us in every data structure.

- Suppose we have a object and we have to traverse it : Traversing of that object will depend on data structure used to store it. ways to traverse



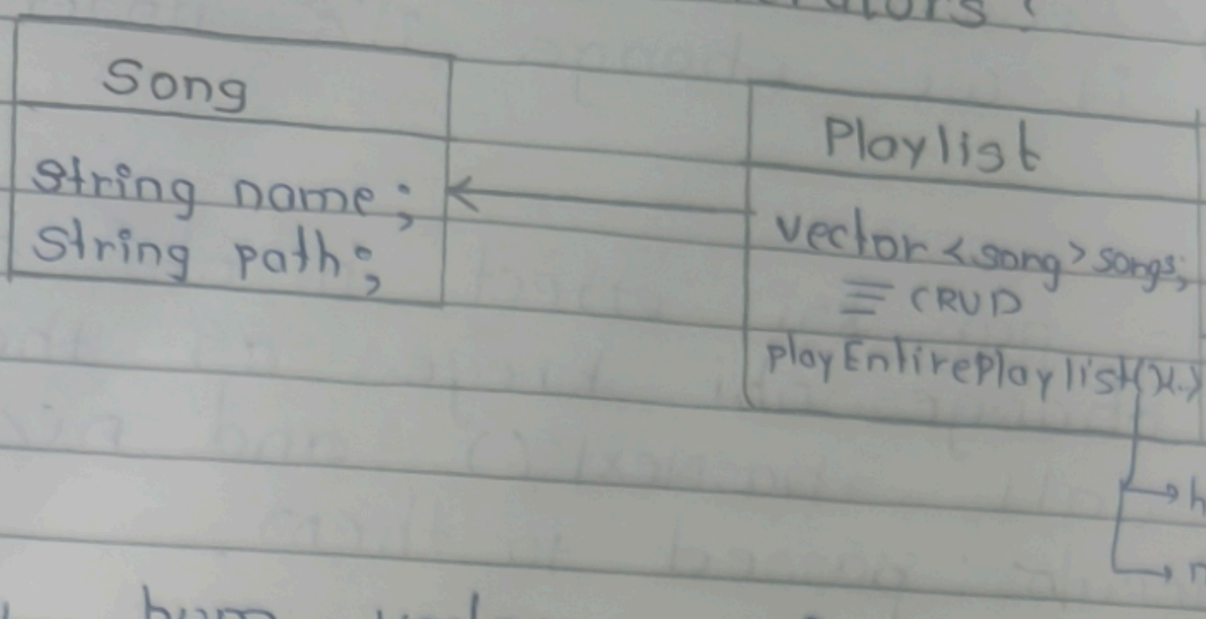
- List of songs can be stored using vector ⇒ traverse using normal for loop ↵

- if stored using L.L then using temp var.

- Along with all these we have another way of traversing known as "iterators"

↳ list.getIterator → it
- ↳ hasNext() → checks if nxt value is there
- ↳ next() → if hasNext() true then moves to next value

Why do we need Iterators?

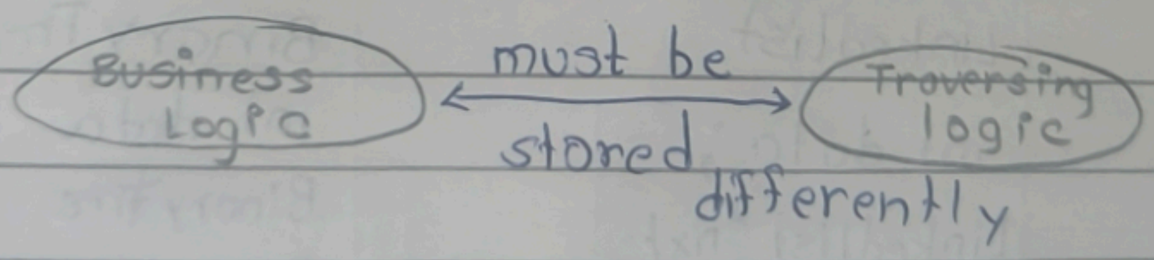


- Initially, hum vector mein store kar rahe the ~~vector~~ songs ko so we can iterate using for loop — for(songs : song)
  song → path

- Later, we want to store in linked list so we need to change the traversing logic as it is different for LL.

- which breaks SRP principle as we are storing both business logic and traversing logic in same class

- ∴



- which means playlist should not know how to traverse, it just call one method and get another song.

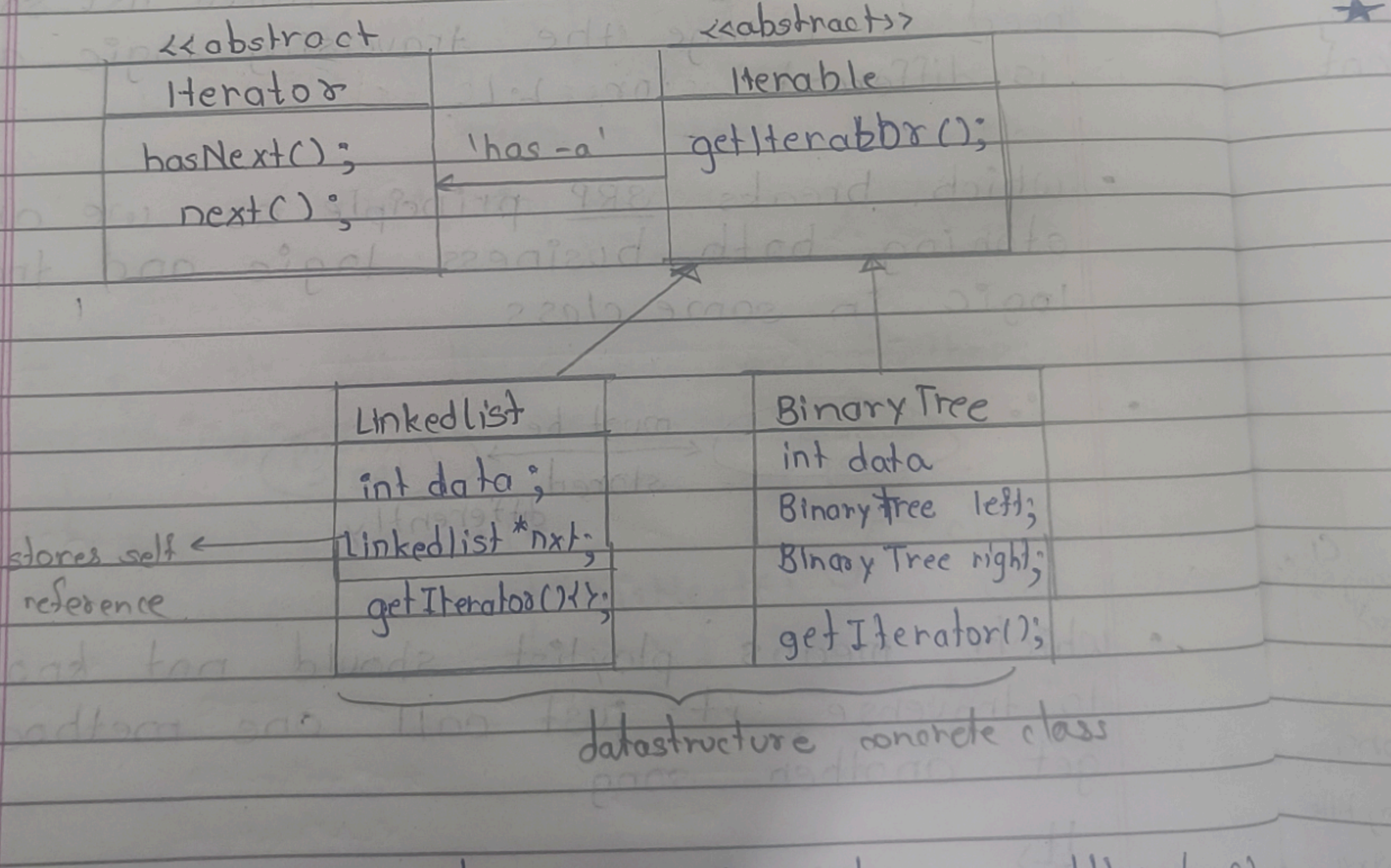
↳ We will use Iterator for that
↓
though more work but to prevent breaking SOLID principles.

- Using Iterator we will be able to follow SRP even if we change it's underlying ds.

- Methods par koi effect hi nhi aayega coz hum changes nhi krege as they will just call hasNext() and next() method of iterator passed to them.

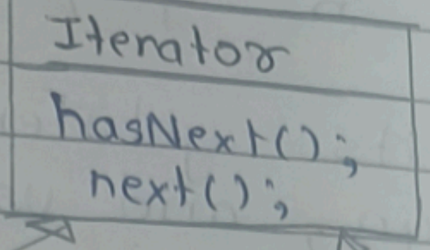
UML Diagram for Playlist example

- First, Iterator class : knows how to iterate particular data structure.



- Ab humne har ds mein baar baar getIterator() method banane ke jagah abstract class banadi - named as Iterable (any ds which can be iteratable should inherit it).

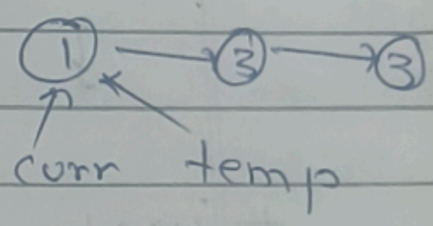
- getIterator() method ↓ humme return karega of every DS ek iterator
  ↳ ~~tut~~ let's create it



LinkedList Iterator
LinkedList * curr
hasNext() {...}
Next() {...}

BinaryTree Iterator
BT * head;
hasNext() {...};
next() {...};

★ HOW L.L traversing happens?




```
hasNext () {
    if (curr -> next != null)
        return true;
    else
        return false;
}
next() {
    curr = curr -> next;
}
```

} this is how it happens

- LinkedList 'has-a' LinkedList Iteraton
- BinaryTree 'has-a' Binary Tree Iterator
- Now, let's assume playlist has songs & it also has its own iteraton.

Page No.
Date

★ How playlist Iterator will work?

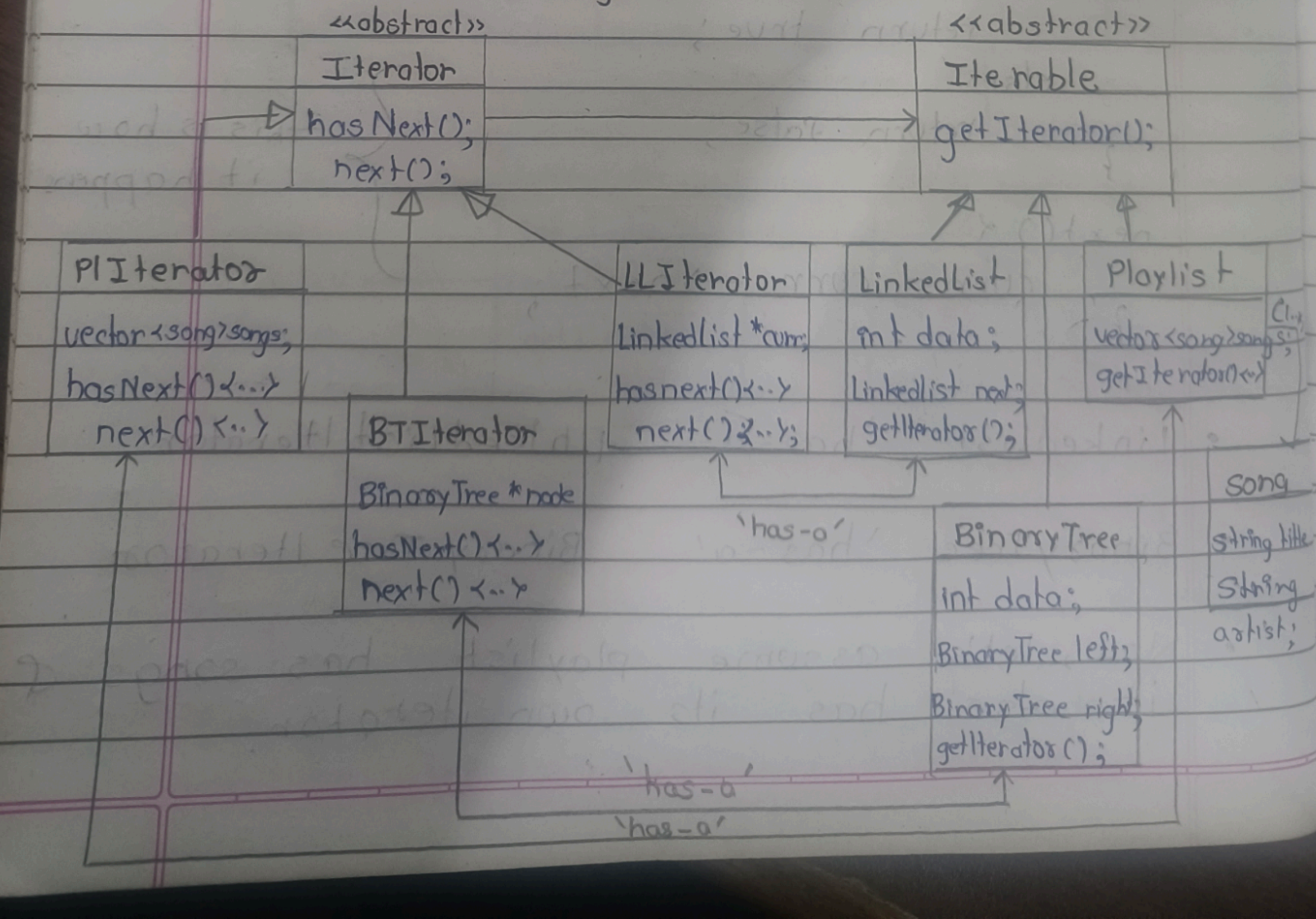
- Client call karega getsongByName() method ko in playlist.

- Playlist call its getIterator() method for traverse karega, and uske baad it gets its playlist iterator

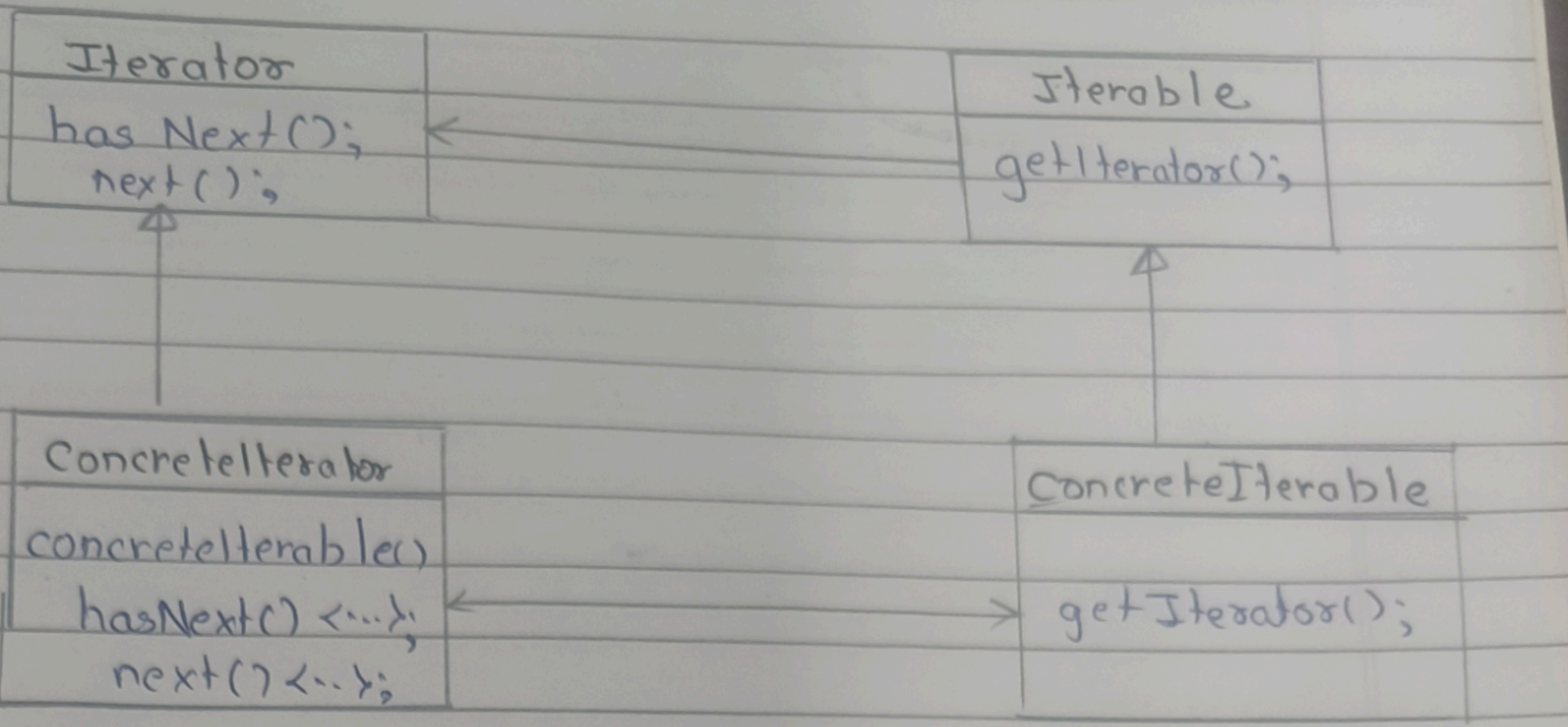
- Fir woh traverse karega playlist ke list of song ko and usse nhi pata hoga ki list stored in vector / linked list, etc

```
while (it -> hasNext()) {
    it -> next()
}
```
} till we find song asked by client

UML final Diagram



standard UML



standard Definition

Iterator provides us a way to access the elements of an aggregate object sequentially without exposing its underlying architecture.