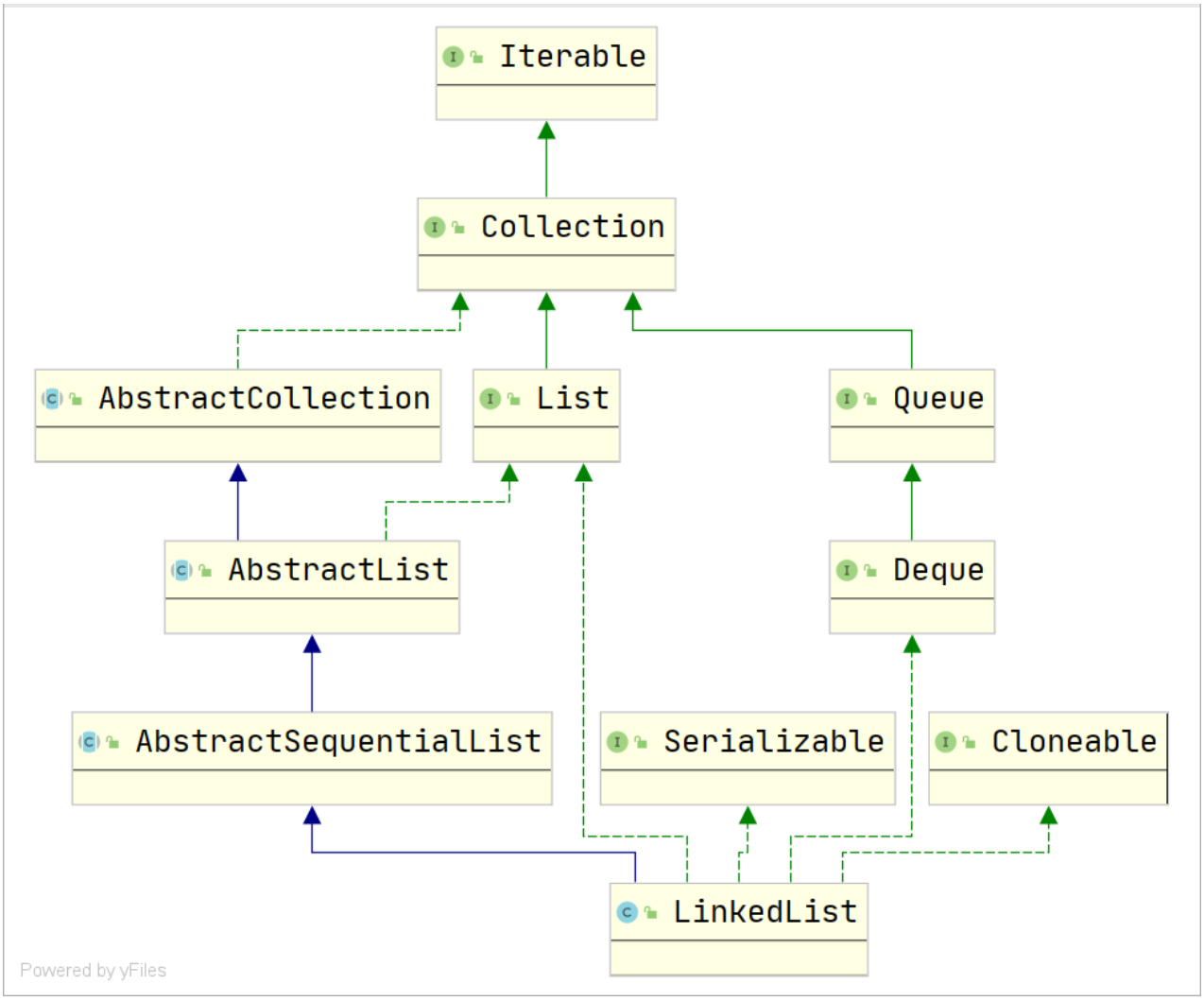
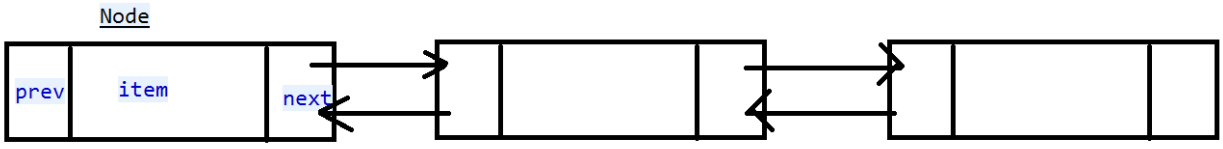
java.util.LinkedList

```
public class LinkedList<E>
extends AbstractSequentialList<E>
implements List<E>, Deque<E>, Cloneable, Serializable
```

`LinkedList` 的直接父类是 `AbstractSequentialList`,实现了 `List` 、 `Deque`



`LinkedList` 是一个双向链表，允许存储 `null` ，此实现不同步（非线程安全的）




```
//实例化
LinkedList list = new LinkedList();

LinkedList list1 = new LinkedList(list);
```

方法名	返回值	描述
addFirst(E e)	void	在该列表开头插入指定的元素
addLast(E e)	void	将指定的元素追加到此列表的末尾/add()
get(int index)	E	返回此列表中指定位置的元素
getFirst()	E	返回此列表中的第一个元素
getLast()	E	返回此列表中的最后一个元素
push(E e)	void	列表的前面插入元素/addFirst()
removeFirst()	E	从此列表中删除并返回第一个元素/poll()/pollFirst()/pop()

- 删除/新增操作效率高

比较 `ArrayList` / `LinkedList`

都实现了 `List` 接口,都是有序的、可以重复的、可以存null值得集合，可以使用下标访问元素

- `ArrayList` 在随机访问（获取元素时）效率比 `LinkedList` 高
  - `ArrayList` 底层实现是数组, 默认容量是 10 ...
  - `LinkedList` 底层实现的链表(双向链表)
- 在添加元素到末尾时，两个集合效率差不多
- 在任意位置添加元素时， `LinkedList` 效率更高
- 在任意位置删除元素时， `LinkedList` 效率高
- 内存， `ArrayList` 使用的是连续空间