Absolutely! As an expert in Java programming, here's a **complete list of all the methods available in the `java.util.Stack` class**, including the ones it inherits from its parent classes (`Vector`, `AbstractList`, `AbstractCollection`, and `Object`).

✅ Stack Class Overview

Java `Stack` is a subclass of `Vector` and extends `Vector<E>`. It uses **LIFO (Last In, First Out)** principle.

📌 Methods Declared in Stack

These are the **specific methods** declared in the `Stack` class:

Method	Description
`boolean empty()`	Checks if the stack is empty.
`E peek()`	Returns the object at the top without removing it.
`E pop()`	Removes and returns the top object of the stack.
`E push(E item)`	Pushes an item onto the top of the stack.
`int search(Object o)`	Returns the 1-based position from the top of the stack if the object exists, else -1.

📌 Important Methods Inherited from Vector

Since `Stack` extends `Vector`, it also inherits **all methods of Vector**. Here are **major Vector methods grouped logically**:

- 🔹 **Element Access & Modification**

Method	Description
`E get(int index)`	Gets the element at the given index.
`E set(int index, E element)`	Replaces the element at index with the given one.
`void addElement(E obj)`	Adds element to the end.
`void insertElementAt(E obj, int index)`	Inserts element at specific position.
`void removeElementAt(int index)`	Removes element at specified index.
`boolean removeElement(Object obj)`	Removes first occurrence of object.
`void setElementAt(E obj, int index)`	Sets object at index.
`E firstElement()`	Returns the first element.

`E lastElement()`	Returns the last element.

- **Capacity & Size**

Method	Description
`int size()`	Returns number of elements.
`int capacity()`	Returns current capacity.
`void ensureCapacity(int minCapacity)`	Ensures capacity is at least `minCapacity`.
`void trimToSize()`	Trims capacity to current size.
`void setSize(int newSize)`	Sets new size, filling with nulls if growing.
`boolean isEmpty()`	Returns true if vector is empty.

- **Search & Query**

Method	Description
`boolean contains(Object o)`	Checks if element exists.
`int indexOf(Object o)`	Returns index of first occurrence.

`int lastIndexOf(Object o)`	Returns index of last occurrence.
`int indexOf(Object o, int index)`	Index from specific position.
`int lastIndexOf(Object o, int index)`	Last index from specific position.
`boolean containsAll(Collection<?> c)`	Checks if vector contains all elements of collection.

◆ **Bulk Operations**

Method	Description
`void addAll(Collection<? extends E> c)`	Adds all elements from collection.
`void addAll(int index, Collection<? extends E> c)`	Inserts all at specific index.
`boolean removeAll(Collection<?> c)`	Removes all matching elements.
`boolean retainAll(Collection<?> c)`	Keeps only elements in collection.
`void clear()`	Removes all elements.

◆ **Conversion & Iteration**

Method	Description
`Object[] toArray()`	Converts vector to array.
`<T> T[] toArray(T[] a)`	Converts to typed array.
`Enumeration<E> elements()`	Returns enumeration of vector.
`Iterator<E> iterator()`	Returns iterator.
`ListIterator<E> listIterator()`	List iterator.
`ListIterator<E> listIterator(int index)`	List iterator from index.
`List<E> subList(int fromIndex, int toIndex)`	Sublist between two indices.

📌 Methods Inherited from java.lang.Object

Method	Description
`boolean equals(Object o)`	Checks equality.

`int hashCode()`	Hash code of object.
`String toString()`	String representation.
`Class<?> getClass()`	Gets class info.
`void notify()`	Wakes up a single thread.
`void notifyAll()`	Wakes all threads.
`void wait()`	Causes current thread to wait.
`void wait(long timeout)`	Waits for specified milliseconds.
`void wait(long timeout, int nanos)`	Waits with nanos.
`Object clone()`	Creates and returns a copy.
`protected void finalize()`	Called by GC before object is destroyed.

✅ Summary

If you're building or solving problems using Java Stack, you'll mostly use:

- `push(E item)`
- `pop()`
- `peek()`
- `empty()`
- `search(Object o)`

But understanding inherited `Vector` methods gives you a **big power boost** for advanced operations.

If you want **Java code examples** for all `Stack` methods, I can provide a complete reference file or project. Would you like that?