Collections class addAll method

Collections class has a addAll method that takes list and varargs as parameter. Varargs can accept array or individual elements.

This method is used to add all the elements of type T into underlying Collection.

It can throw few exceptions if conditions are not met

UnsupportedOperationException – if the collection does not support add operation. Example if Collection is read-only and we try to use this method will result in UnsupportedOperationException.

NullPointerException – if the collection does not support null values then it will throw NullPointerException.

IllegalArgumentException – if some property of value in elements does not allow to being added to collection.

It returns Boolean which indicates whether all the elements are inserted or not.

Below program demonstrates the usage of addAll(..) method

**package** org.collections;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.List;

**public** **class** CollectionClass {

**public** **static** **void** main(String[] args) {

*collectionsAddAllDemo*();

}

**public** **static** **void** collectionsAddAllDemo() {

String[] elements = { "MySql", "SqlLite", "MongoDB", "Neo4j", "Oracle" };

List<String> languages = *populateLanguages*();

/\*\*

\* Using List<String> and String[]

\* \*/

Collections.*addAll*(languages, elements);

System.***out***.println(languages);

/\*\*

\* Using List<String> and varargs

\* \*/

Collections.*addAll*(languages, "MS Excel", "MS Powerpoint");

System.***out***.println(languages);

}

**public** **static** List<String> populateLanguages() {

List<String> languages = **new** ArrayList<String>();

languages.add("Java");

languages.add("JavaScript");

languages.add("C#");

languages.add("Python");

**return** languages;

}

}

Output

[Java, JavaScript, C#, Python, MySql, SqlLite, MongoDB, Neo4j, Oracle]

[Java, JavaScript, C#, Python, MySql, SqlLite, MongoDB, Neo4j, Oracle, MS Excel, MS Powerpoint]