



SF STATE

SAN FRANCISCO STATE UNIVERSITY
COMPUTER SCIENCE DEPARTMENT

GOOGLE GUAVA

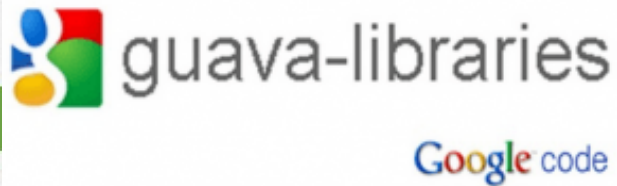
CODE LIBRARIES FOR JAVA AND ANDROID

DUC TA



GOOGLE GUAVA

CODE LIBRARIES FOR JAVA AND ANDROID



Guava Multimap

- **Google Guava** is an open-source set of common libraries for Java, mainly developed by Google engineers. (https://en.wikipedia.org/wiki/Google_Guava)
1. Basic utilities to reduce menial labors to implement common methods and behaviors.
 2. An extension to the Java collections framework (JCF) formerly called the **Google Collections Library**.
 3. Other utilities which provide convenient and productive features such as functional programming, graphs, caching, range objects, and hashing.

Google Guava

Original author(s)	Kevin Bourrillion and Jared Levy (Google Collections Library) ^[1]
Developer(s)	Google
Stable release	24.0 / January 31, 2018 ^[2]
Repository	https://github.com/google/guava 
Development status	Active
Written in	Java
Operating system	Cross-platform
Type	Utility and Collection Libraries
License	Apache License 2.0
Website	github.com/google/guava 

Guava Multimap

- Interface Multimap: <https://google.github.io/guava/releases/snapshot-jre/api/docs/com/google/common/collect/Multimap.html>
- Repository:
 - <https://github.com/google/guava>
 - <https://search.maven.org/#search|gav|1|g:%22com.google.guava%22%20AND%20a:%22guava%22>
- **Add to NetBeans:**
 - Right click on a project
 - Go to Properties → Libraries → Compile → Add JAR/Folder
 - Navigate to the JAR file downloaded from Guava: E.g. “*guava-24.1.jre.jar*”
- **Import**

```
import com.google.common.collect.ArrayListMultimap;  
import com.google.common.collect.Multimap;
```



GOOGLE GUAVA AND TREES

Google Guava TreeTraverser

- URL: <https://github.com/google/guava/blob/master/guava/src/com/google/common/hash/HashCode.java>

com.google.common.collect

Class TreeTraverser<T>

java.lang.Object

com.google.common.collect.TreeTraverser<T>

Direct Known Subclasses:

BinaryTreeTraverser

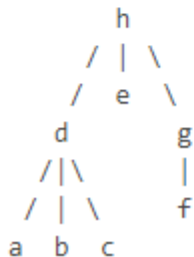
@Beta

@GwtCompatible

public abstract class TreeTraverser<T>
extends Object

Views elements of a type T as nodes in a tree structure.

For example, the tree



Method Summary

All Methods	Static Methods	Instance Methods	Abstract Methods	Concrete Methods
Modifier and Type		Method and Description		
FluentIterable<T>		breadthFirstTraversal(T root)		Returns an unmodifiable iterable over the nodes in a tree structure, using breadth-first traversal.
abstract Iterable<T>		children(T root)		Returns the children of the specified node.
FluentIterable<T>		postOrderTraversal(T root)		Returns an unmodifiable iterable over the nodes in a tree structure, using post-order traversal.
FluentIterable<T>		preOrderTraversal(T root)		Returns an unmodifiable iterable over the nodes in a tree structure, using pre-order traversal.
static <T> TreeTraverser<T>		using(Function<T,? extends Iterable<T>> nodeToChildrenFunction)		Returns a tree traverser that uses the given function to navigate from a node to its children.

Google Guava TreeTraverser Source Code

- URL: <https://github.com/google/guava/blob/master/guava/src/com/google/common/collect/TreeTraverser.java>

The screenshot shows the GitHub interface for the file `TreeTraverser.java` in the `google/guava` repository. The repository has 2,164 watches, 24,005 stars, and 5,493 forks. The file is located at `guava / guava / src / com / google / common / collect / TreeTraverser.java`. A recent commit by `cgdecker` is shown, titled "Push back some Guava removal dates to July 2018." The file is 291 lines long (262 sloc) and 8.73 KB. The code is displayed with line numbers and includes a license header.

```

1  /*
2   * Copyright (C) 2012 The Guava Authors
3   *
4   * Licensed under the Apache License, Version 2.0 (the "License");
5   * you may not use this file except in compliance with the License.
6   * You may obtain a copy of the License at
7   *
8   * http://www.apache.org/licenses/LICENSE-2.0
9   *
10  * Unless required by applicable law or agreed to in writing, software
11  * distributed under the License is distributed on an "AS IS" BASIS,
12  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13  * See the License for the specific language governing permissions and
14  * limitations under the License.
15  */
16
17  package com.google.common.collect;

```

Use TreeTraverser

- URL: <https://static.javadoc.io/com.google.guava/guava/18.0/com/google/common/collect/class-use/TreeTraverser.html>

Uses of Class

com.google.common.collect.TreeTraverser

Packages that use TreeTraverser

Package	Description
com.google.common.collect	This package contains generic collection interfaces and implementations, and other utilities for working with collections.
com.google.common.io	This package contains utility methods and classes for working with Java I/O; for example input streams, output streams, readers, writers, and files.

Uses of TreeTraverser in com.google.common.collect

Subclasses of TreeTraverser in com.google.common.collect

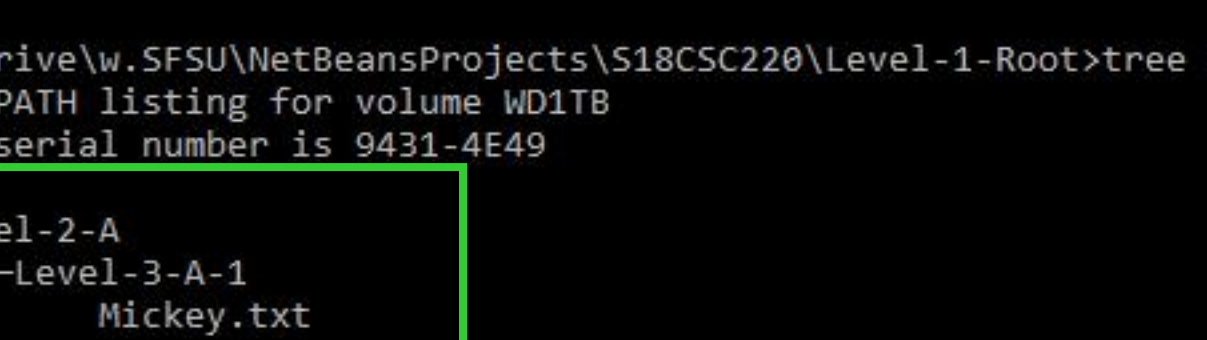
Modifier and Type	Class and Description
class	BinaryTreeTraverser<T> A variant of TreeTraverser for binary trees, providing additional traversals specific to binary trees.

Uses of TreeTraverser in com.google.common.io

Methods in com.google.common.io that return TreeTraverser

Modifier and Type	Method and Description
static TreeTraverser<File>	Files.fileTreeTraverser() Returns a TreeTraverser instance for File trees.

Analyze the **DirListing** program line-by-line and do more experiments.

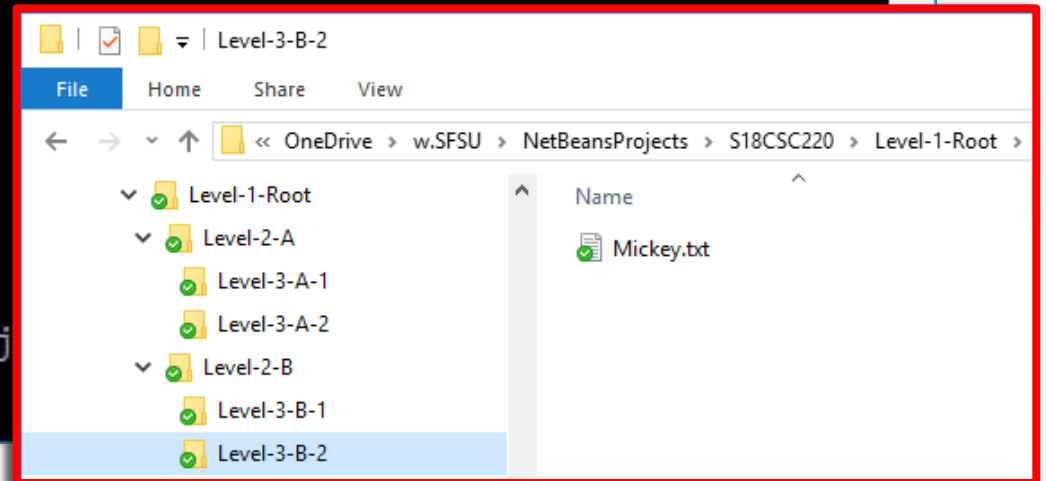


The screenshot displays a Windows Command Prompt window and a File Explorer window. The Command Prompt shows the execution of the 'tree /F' command in the directory D:\OneDrive\w.SFSU\NetBeansProjects\S18CSC220\Level-1-Root. The output lists the folder structure, including Level-2-A, Level-2-B, and their respective Level-3 subfolders, each containing a Mickey.txt file. The File Explorer window shows the same directory structure visually, with the 'Level-3-B-2' folder selected.

```
D:\OneDrive\w.SFSU\NetBeansProjects\S18CSC220\Level-1-Root>tree /F
Folder PATH listing for volume WD1TB
Volume serial number is 9431-4E49
D:.\
├── Level-2-A
│   ├── Level-3-A-1
│   │   └── Mickey.txt
│   └── Level-3-A-2
│       └── Mickey.txt
└── Level-2-B
    ├── Level-3-B-1
    │   └── Mickey.txt
    └── Level-3-B-2
        └── Mickey.txt
```

File Explorer path: < OneDrive > w.SFSU > NetBeansProjects > S18CSC220 > Level-1-Root

- Level-1-Root
 - Level-2-A
 - Level-3-A-1 (Mickey.txt)
 - Level-3-A-2 (Mickey.txt)
 - Level-2-B
 - Level-3-B-1 (Mickey.txt)
 - Level-3-B-2 (Mickey.txt)



GOOGLE GUAVA AND TREES

Analyze the **DirListing.java** program line-by-line and do more experiments.

The image displays two windows side-by-side, illustrating a directory tree structure.

Left Window (Command Prompt): The title bar reads "Select Command Prompt". The command prompt shows the following output:

```
D:\OneDrive\w.sfsu\NetBeansProjects\S18CSC220\Level-1-Root-BigTree> dir
Folder PATH listing for volume WD1TB
Volume serial number is 9431-4E49
D:
.
├── Level-2-A
│   ├── Level-3-A-1
│   │   ├── Level-4-A
│   │   │   ├── Level-5-A
│   │   │   │   ├── Level-6-A
│   │   │   │   │   ├── Level-7-A
│   │   │   │   │   │   ├── Level-8-A
│   │   │   │   │   │   │   ├── Level-9-A
│   │   │   │   │   │   │   │   Mickey.txt
│   │   │   │   │   │   │   └──
│   │   │   │   │   │   └──
│   │   │   │   │   └──
│   │   │   │   └──
│   │   │   └──
│   │   └──
│   └──
├── Level-3-A-2
│   └── Mickey.txt
├── Level-2-B
│   ├── Level-3-B-1
│   │   └── Mickey.txt
│   └── Level-3-B-2
│       └── Mickey.txt
├── Level-2-C
│   ├── Level-3-C-1
│   │   └── Mickey.txt
│   └── Level-3-C-2
│       └── Mickey.txt
├── Level-2-D
│   ├── Level-3-D-1
│   │   └── Mickey.txt
│   └── Level-3-D-2
│       └── Mickey.txt
└── Level-2-E
    ├── Level-3-E-1
    │   └── Mickey.txt
    └── Level-3-E-2
        └── Mickey.txt
```

Right Window (File Explorer): The title bar reads "Level-3-E-2". The address bar shows "Level-3-E-2". The left pane shows a tree view of the directory structure, with "Level-3-E-2" selected. The right pane shows the contents of the selected folder, which is "Mickey.txt".



GOOGLE GUAVA AND GRAPHS

Google Guava Graphs

- URL: <https://github.com/google/guava/wiki/GraphsExplained>
- Graph Types: **Graph** | **ValueGraph** | **Network** ← sibling types, none is a subtype of any of the others
- Graph Types: **MutableGraph/ImmutableGraph** | **Mutable/ImmutableValueGraph** | **Mutable/ImmutableNetwork**
- Building Graph Instances: **GraphBuilder** | **ValueGraphBuilder** | **NetworkBuilder** others

Graphs, Explained

► Pages 66

Guava's `common.graph` is a library for modeling `graph`-structured data, that is, entities and the relationships between them. Examples include webpages and hyperlinks; scientists and the papers that they write; airports and the routes between them; and people and their family ties (family trees). Its purpose is to provide a common and extensible language for working with such data.

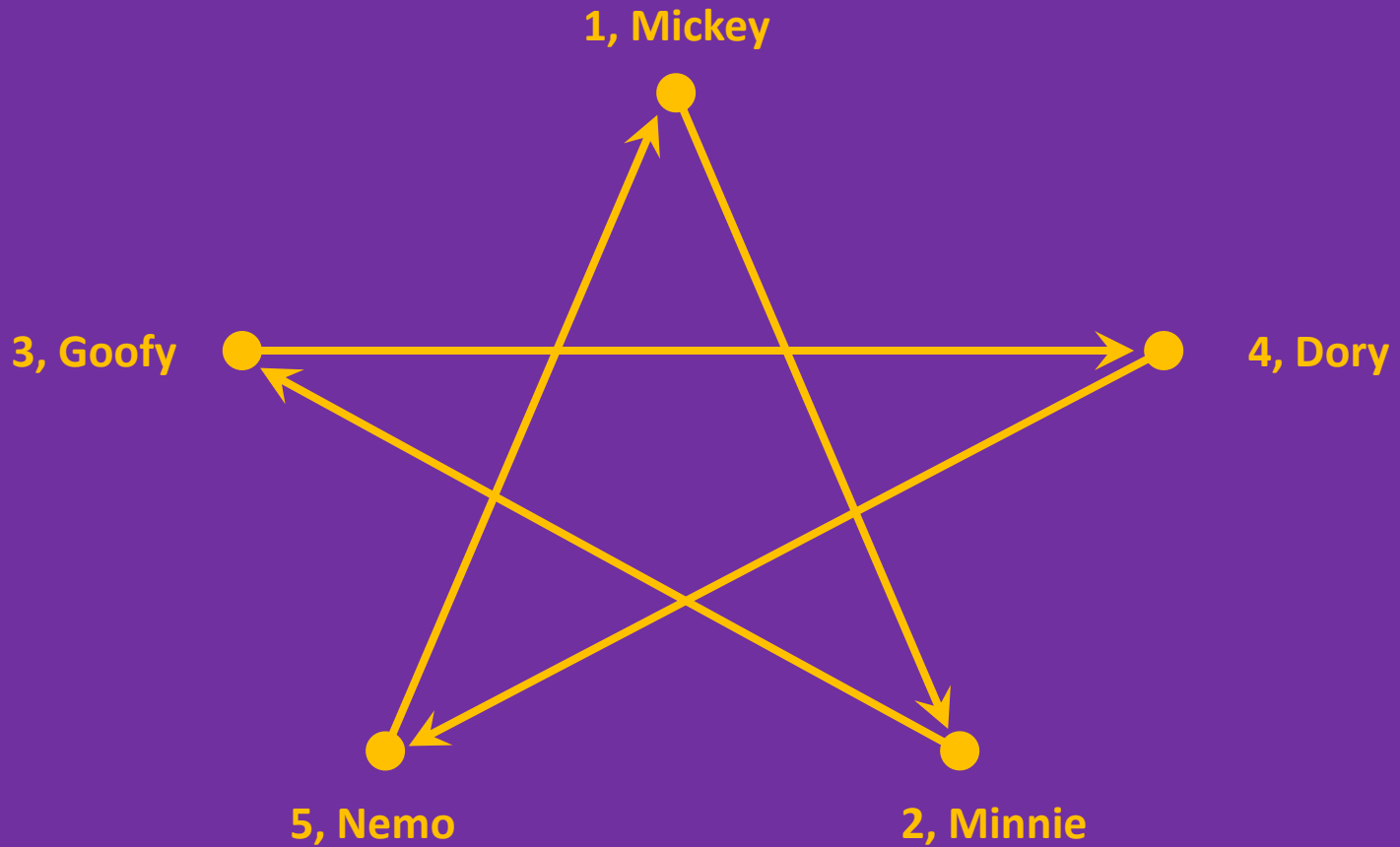
Definitions

A graph consists of a set of **nodes** (also called vertices) and a set of **edges** (also called links, or arcs); each edge connects nodes to each other. The nodes incident to an edge are called its **endpoints**.

(While we introduce an interface called `Graph` below, we will use "graph" (lower case "g") as a general term referring to this type of data structure. When we want to refer to a specific type in this library, we capitalize it.)

- [Introduction](#)
- [Basic Utilities](#)
 - [Using/avoiding null](#)
 - [Optional](#)
 - [Preconditions](#)
 - [Ordering](#)
 - [Creation](#)
 - [Chaining](#)
 - [Application](#)
 - [Object methods](#)
 - [equals](#)
 - [hashCode](#)
 - [toString](#)
 - [compare/compareTo](#)
 - [Throwables](#)

Analyze the **DisneyStar.java** program line-by-line and do more experiments.



GOOGLE GUAVA AND GRAPHS

More Libraries: **JUNG**

<http://jung.sourceforge.net/>


JGraphT <http://jgrapht.org/>

JUNG

Java Universal Network/Graph Framework

- Overview
- Download
- Documentation
- Examples
- Wiki
- Projects Using JUNG
- FAQ
- Support
- Team
- Presentations
- Bug Tracker
- Sourceforge
- Acknowledgements

Links

 SOURCEFORGE

Overview

JUNG — the Java Universal Network/Graph Framework—is a software library that provides a common and extendible language for the modeling, analysis, and visualization of data that can be represented as a graph or network. It is written in Java, which allows JUNG-based applications to make use of the extensive built-in capabilities of the Java API, as well as those of other existing third-party Java libraries.

The JUNG architecture is designed to support a variety of representations of entities and their relations, such as directed and undirected graphs, multi-modal graphs, graphs with parallel edges, and hypergraphs. It provides a mechanism for annotating graphs, entities, and relations with metadata. This facilitates the creation of analytic tools for complex data sets that can examine the relations between entities as well as the metadata attached to each entity and relation.

The current distribution of JUNG is available on SourceForge. The current distribution of JUNG is available on SourceForge. The current distribution of JUNG is available on SourceForge.

JUNG also provides a set of APIs for network analysis, such as shortest path calculation, centrality calculation, and community detection. Users can use one of these APIs to analyze a graph. The mechanisms are provided in the JUNG API.

As an open-source project, JUNG will make it easier for you to contribute to the project. You can continually re-invent the wheel.

— The JUNG Framework

JGraphT

[Open Hub](#) **26 Developers**

Latest News

Release 1.1.0 is now available, dual-licensed under LGPL [and](#) EPL.

Once upon a time, JGraphT source code lived in an svn repository, but these days it lives in [github](#). Please send all new contributions there in the form of pull requests!

If you enjoy using JGraphT, please click the Like button for our [Facebook page](#)!



JGraphT
638 likes

[Like Page](#) [Learn More](#)

Be the first of your friends to like this

See you next class!