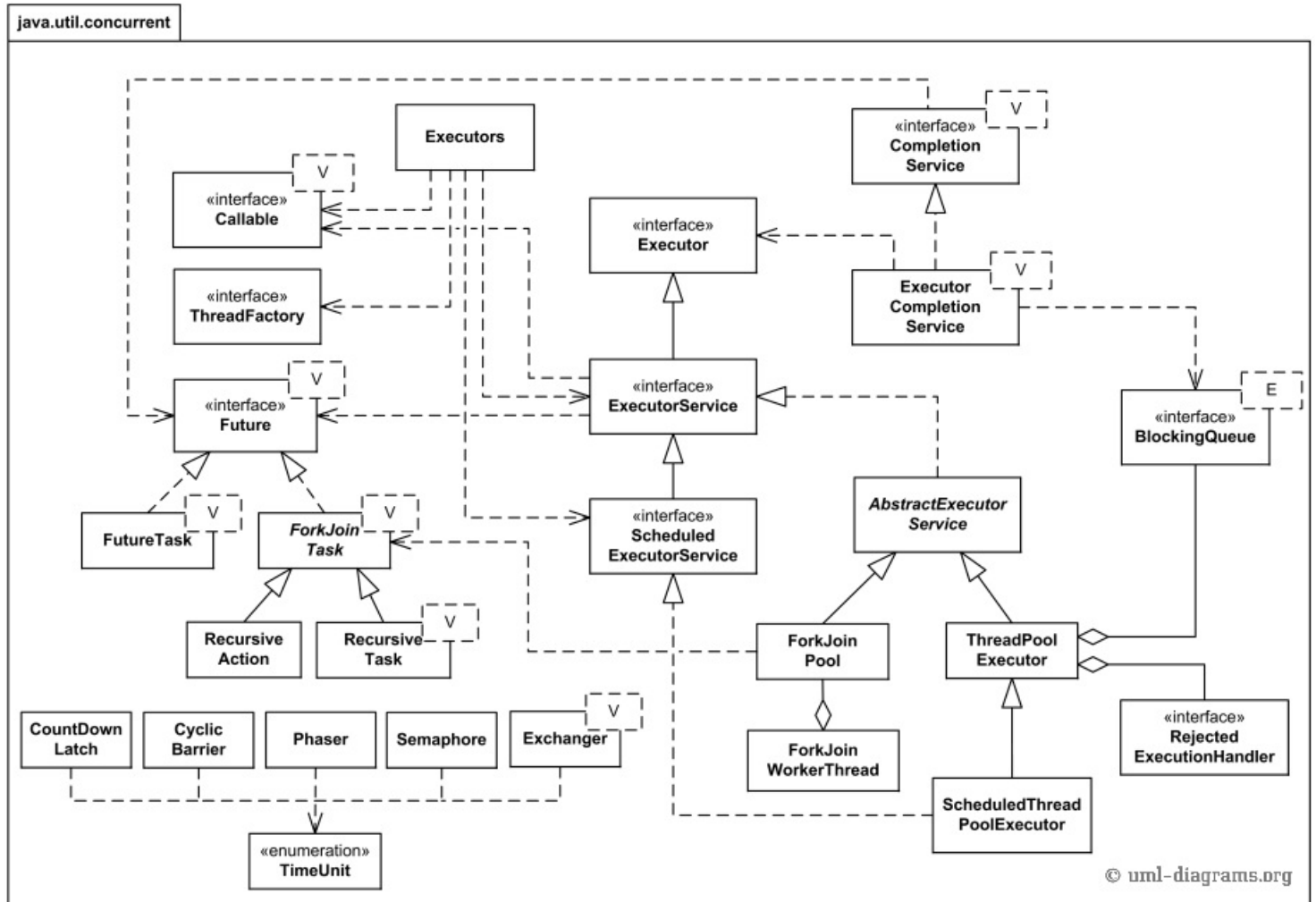


# Java™ 7 util.concurrent API

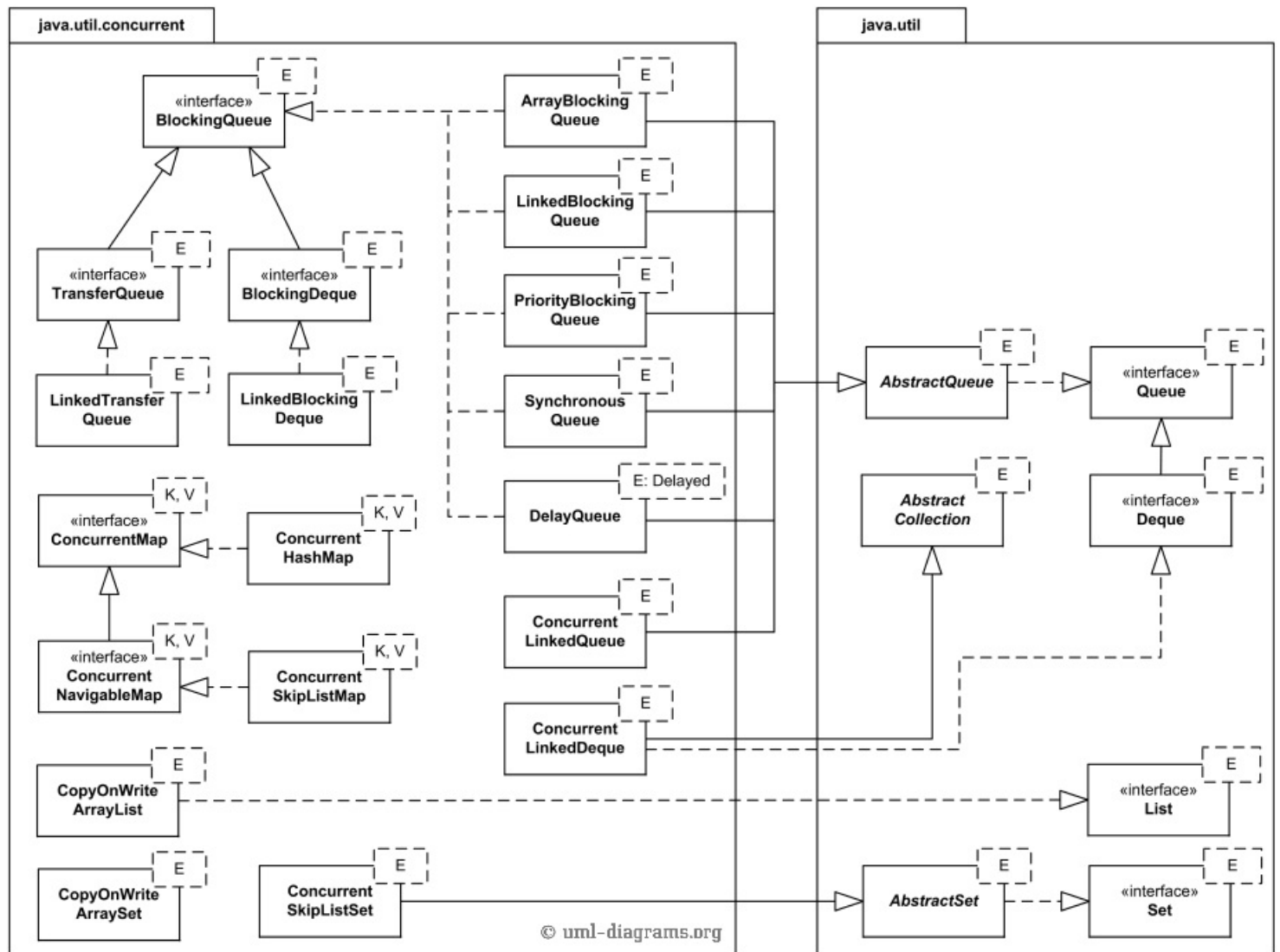
## UML Class Diagram Examples

Here we provide several UML **class diagrams** for the Java™ 7 **java.util.concurrent** package. Several java.util.concurrent.\* packages introduced with version 5.0 of the Java platform added high-level concurrency features to the Java and new concurrent data structures to the Java Collections Framework.



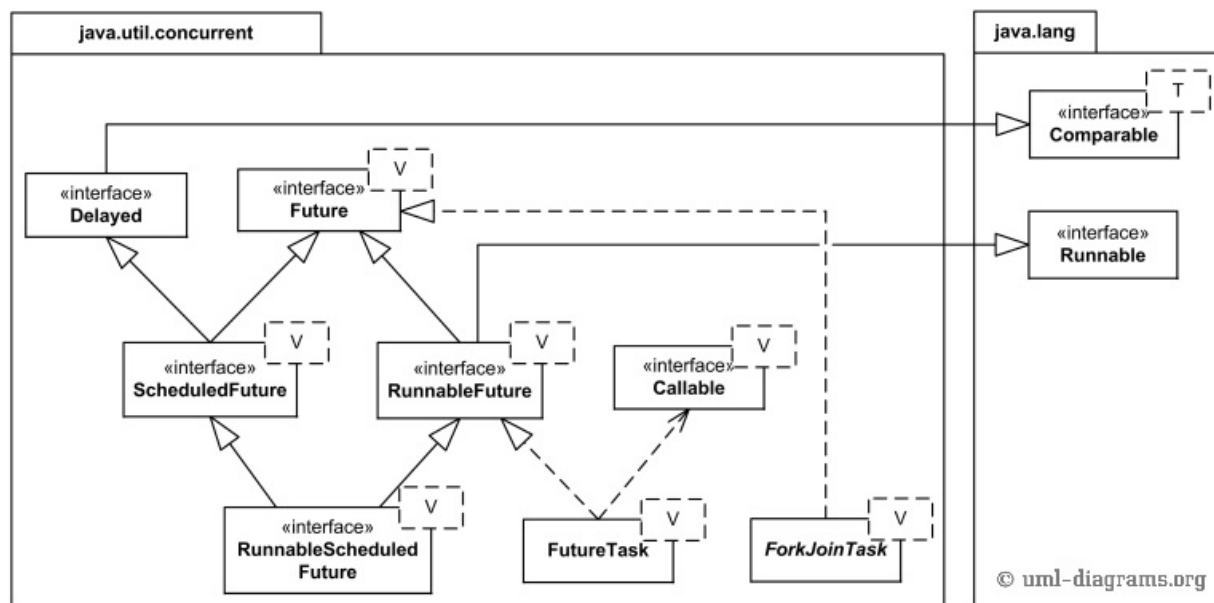
UML class diagram for the Java™ 7 executors and thread pool managers from the java.util.concurrent package.

**Executors** define a high-level API for launching and managing threads to support large-scale applications mostly by adding **thread pool management** abilities. The java.util.concurrent package includes several thread pool management implementation classes.



UML class diagram for concurrent collections from the Java 7 `java.util.concurrent` package.

**Concurrent collections** are also part of the `java.util.concurrent` package. These collections reduce the need for synchronization and are designed to support concurrent access and modifications of the large collections of data.



UML class diagram for asynchronous results (futures) from the Java 7 `java.util.concurrent` package.

The **Future<V>** interface represents the result of an asynchronous computation, where type `V` is the result type returned by the `Future`'s `get` method. Methods of this interface allow to wait for the computation to complete, to cancel execution of the task, to check if the computation is complete or was cancelled, and to retrieve the result of the computation.

The **Delayed** interface allows to mark objects that should be acted upon after a given delay. `ScheduledFuture<V>` interface extends both `Future<V>` and `Delayed`, and is usually a result of scheduling a task with a `ScheduledExecutorService`.

The **FutureTask** class is an implementation of `Future` that implements `java.lang.Runnable` as required by `RunnableFuture` interface, and thus may be executed by an `Executor`.

*Noticed a spelling error? Select the text using the mouse and press Ctrl + Enter.*

Follow [@uml\\_diagrams](#)      by [Kirill Fakhroutdinov](#)

This document describes **UML 2.5** and is based on **OMG™ Unified Modeling Language™ (OMG UML®) 2.5** specification **[UML 2.5 FTF - Beta 1]**.

All UML diagrams were created in **Microsoft Visio** 2007-2016 using **UML 2.2 stencils**. You can send your comments and suggestions to [webmaster](#) at [webmaster@uml-diagrams.org](mailto:webmaster@uml-diagrams.org).

Copyright © 2009-2018 [uml-diagrams.org](http://uml-diagrams.org). All rights reserved.