Версионная память. Формулировка задачи.

Основной задачей данной работы является создание структур данных, поддерживающих версионную память. Эти структуры работают следующим образом: для каждого потока, который пользуется ими для записи, создается своя версия структуры. Все полученные версии затем объединяются при помощи алгоритма слияния. Особенностью такой памяти является то, что результат работы алгоритма слияния должен быть детерминирован, т.е. не зависеть от того, как распределяются ресурсы компьютера между потоками.

Планируется реализовать следующие структуры данных: версионную переменную, стек, очередь и сбалансированное дерево. Для всех этих структур будут реализованы специальные алгоритмы слияния, которые позволят сохранять их основные свойства. Например, сбалансированное дерево остается сбалансированным.

Так же будет разработан пользовательский механизм создания стратегий объединения версий.

Этапы разработки системы

Данную систему планируется реализовать на языке Scala.

# Версионная память

## Цель

Реализовать структуры данных с поддержкой одновременного многопоточного независимого изменения с возможностью слияния версий и детерминированным результатом вычислений. Среди реализованных структур должны быть такие структуры, как стек, очередь и множество. Пользователю должна быть предоставлена возможность определения своих стратегий слияния для реализованных структур.

## Принцип работы версионной памяти

Вместо блокировки объекта, при его изменении создается новая копия этого объекта для текущего треда. Все копии объекта хранятся в древовидной структуре. После окончания работы всех тредов, выполняется слияние копий объекта. Microsoft предлагает решение, в котором используются блокировки внутри версии.

## Реализация

Языком для реализации данного проекта выбран Scala. Выбор сделан по двум причинам: во-первых, Scala поддерживает как функциональную, так и объектно-ориентированную парадигму, что обеспечит дополнительную гибкость при написании проекта, во-вторых, этот язык