**Версионная память**

**Описание реализации (язык С#)**

* Каждая версионная структура данных содержит дерево своих версий для каждого потока.
* Узел дерева содержит:
  + Идентификатор потока (по сути идентификатор версии)
  + Тип узла (Root/Fork/Join/Edit)
  + Копию исходной структуры данных (копия рабочей копии родительского узла)
  + Рабочую копию структуры данных
  + Историю изменений структуры (время, тип изменения, значение элемента структуры данных) относительно родительского узла
  + Список дочерних узлов
* Версия структуры данных потока – путь от корня дерева версий до листа с идентификатором потока. Рабочими для каждого потока являются только узлы дерева (за исключением слияния). Как только у узла появляется потомок, то узел не изменяется до момента слияния.
* Слияние двух версий происходит только в том случае, когда оба листа достижимы из узла типа Fork. При слиянии образуется узел типа Join, где исходная структура та же, что и в узле Fork, рабочая копия структуры результат слияния, история изменений – результат применения стратегии слияния.
* Для потока определяется метод void Run(Collection[]) принимает структуры данных, изменяемые в потоке и создает для них версию (иначе начальное состояние структуры данных для новой версии не детерминировано, что может повлечь недетерминированность алгоритма слияния).
* Для структуры определяется методы
  + bool CanJoin(ThreadId) – для проверки возможности слияния текущей версии и версии ThreadId (возможно произвести только слияние версий)
  + void Join(ThreadId, MergeVersionStrategy, MergeValueStrategy),  
    который принимает идентификатор потокf и производит слияние версий используя либо преданные в качестве параметров пользовательские стратегии слияния, либо стратегии по умолчанию.
* Стратегия слияния версий структуры данных – метод принимающий набор операций от момента запуска потока до момента слияния двух версий. Возвращает набор операций, которые будут применены к состоянию структуры данных на момент запуска потока. (результат применения – новая версия структуры данных)
* Стратегия слияния версий значений коллекции – метод принимающий два значения из разных версий коллекции. Возвращает результат слияния.
* Стратегия по умолчанию обеспечивает детерминированное слияние версий.
* Для Стека, Очереди и Множества допускается, что элементы не изменяются, пока находятся в коллекции.
* Для дерева отсутствуют стратегии по умолчанию, однако реализованы

**Распределение задач**

* Булдаков Иван
  + Разработка и реализация версионных структур данных. Реализация дерева версий.
* Емельянова Татьяна
  + Разработка дерева версий. Разработка и реализация алгоритмов слияния обеспечивающих детерминированность и механизма определения пользовательских алгоритмов слияния.