



Библиотека для интервальных вычислений JInterval: принципы организации

Данилов М.В., Дронов К.С., Жилин С.И., Тепикин Е.Н.
АлтГУ, г. Барнаул

План

- Мотивация
- Принципы проекта JInterval
- Структура библиотеки JInterval
- JInterval с точки зрения пользователя
- Сравнительные тесты производительности
- Итоги

[Мотивация]

- Научные вычисления на Java: за и против
- Java-продукты для интервальных вычислений

Мотивация

- Научные вычисления на Java: за и против
 - За
 - Переносимость Java Virtual Machine (JVM)
 - Безопасное управление памятью
 - Строгая модель безопасности
 - Встроенная поддержка сети
 - Параллельные и распределенные вычисления (потoki, RMI)
 - Стандартные интерфейсы для графики, GUI, БД
 - Широкое распространение

-
1. K.A. Ritley *Java as Scientific Programming Language* // Developer.com, 2001.
 2. J.W. von Gudenberg *OOP and Interval Arithmetic – Language Support and Libraries* // Numerical Software with Result Verification, LNCS 2991/2004, 2004, 467-503.

Мотивация

- Научные вычисления на Java: за и против
 - **Против**
 - Невысокая производительность
 - Виртуальная машина
 - Медленные интерпретаторы
 - Накладные расходы на безопасный доступ к памяти
 - Ограничения языка
 - Нет примитивного типа «структура»
 - Нет перегрузки операторов
 - Нет истинно многомерных массивов
 - Неполная поддержка IEEE 754
 - Недостаток научного ПО, библиотек на Java

-
1. K.A. Ritley *Java as Scientific Programming Language* // Developer.com, 2001.
 2. J.W. von Gudenberg *OOP and Interval Arithmetic – Language Support and Libraries* // Numerical Software with Result Verification, LNCS 2991/2004, 2004, 467-503.

Мотивация

- Интервальные Java-библиотеки
 - **IA_math**, 1997
 - Timothy J. Hickey (Brandeis University, Boston, USA)
 - <http://interval.sourceforge.net/interval/>
 - **Java-XSC**, 1999
 - Benjamin R.C. Bedregal (Universidade Federal do Rio Grande do Norte, Natal, Brazil)
 - <http://www.dimap.ufrn.br/~java-xsc/>
 - **Java-XSC**, 2004
 - Marcilia A. Campos (Universidade Federal de Pernambuco, Recife, Brazil)
 - <http://www.cin.ufpe.br/~javaxsc>

Мотивация

- Интервальные Java-библиотеки:
функционал

	IA_math	Java-XSC (Bedregal)	Java-XSC (Campos)
Базовый тип	double	double	double
Классическая ИА	+	+	+
Комплексная ИА	—	+	—
Элементарные функции	+	+	+
Векторы и матрицы	—	+	+
Статистические функции	—	—	+

[Проект JInterval]

- Цель
 - Разработка открытой полнофункциональной Java-библиотеки для интервальных вычислений
- Дата старта
 - 1 сентября 2008
- Участники
 - Данилов Максим Валерьевич, 4 курс МФ АлтГУ
 - Дронов Кирилл Сергеевич, 4 курс МФ АлтГУ
 - Тепикин Егор Николаевич, 4 курс МФ АлтГУ
 - Жилин Сергей Иванович, МФ АлтГУ

[Проект JInterval]

- Принципы
 - Открытость проекта
 - Расширенная функциональность
 - Обеспечение гибкости в выборе приоритетов вычислений – высокая скорость или высокая точность

Проект JInterval

■ Принципы

- Открытость
 - Открытость исходных кодов
 - Открытость для новых участников проекта
- Расширенная функциональность
- Обеспечение гибкости в выборе приоритетов вычислений – высокая скорость или высокая точность

Проект JInterval

■ Принципы

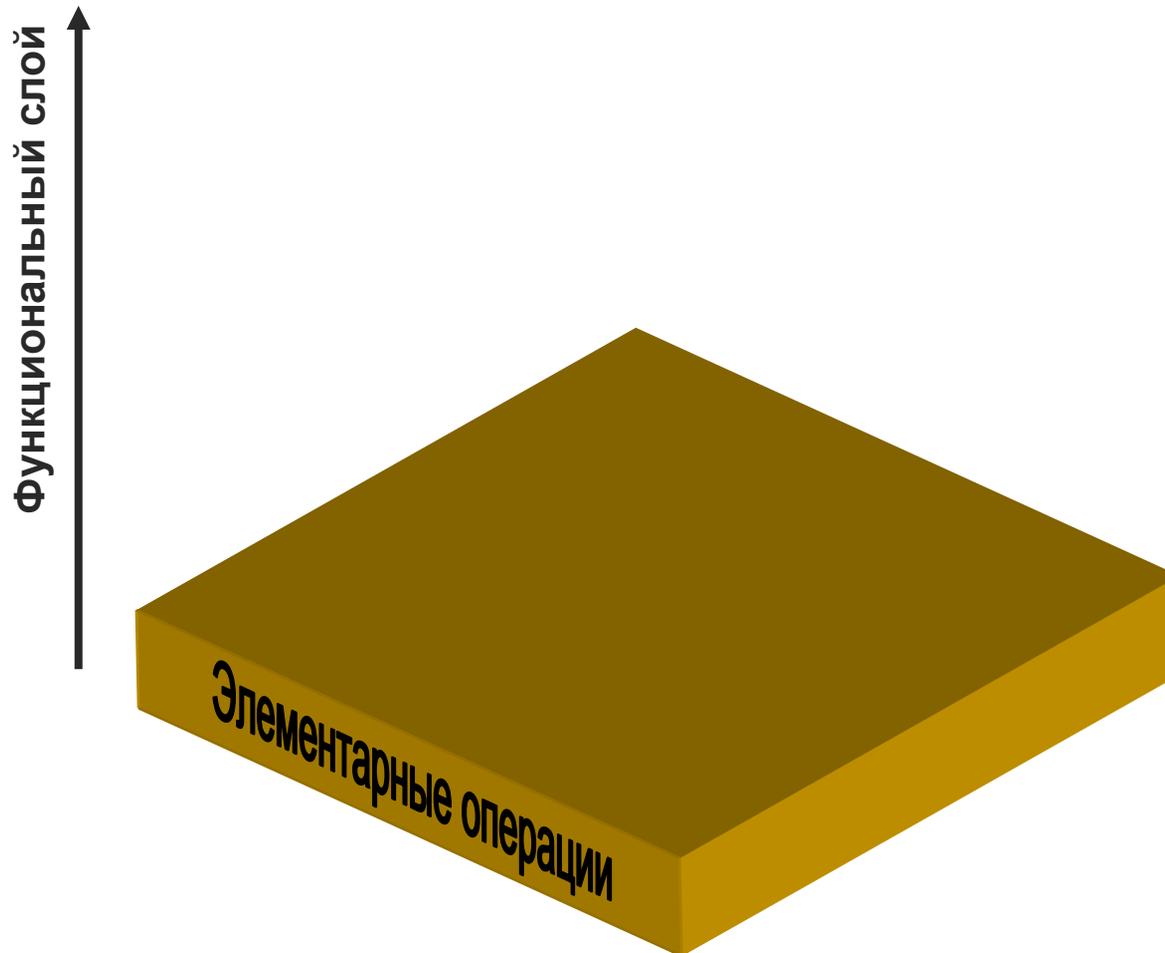
- Открытость
- Расширенная функциональность
 - Различные интервальные арифметики
 - Различные базовые типы для границ интервалов
 - Различные режимы округления
 - Высокоуровневые методы интервального анализа
- Обеспечение гибкости в выборе приоритетов вычислений – высокая скорость или высокая точность

Проект JInterval

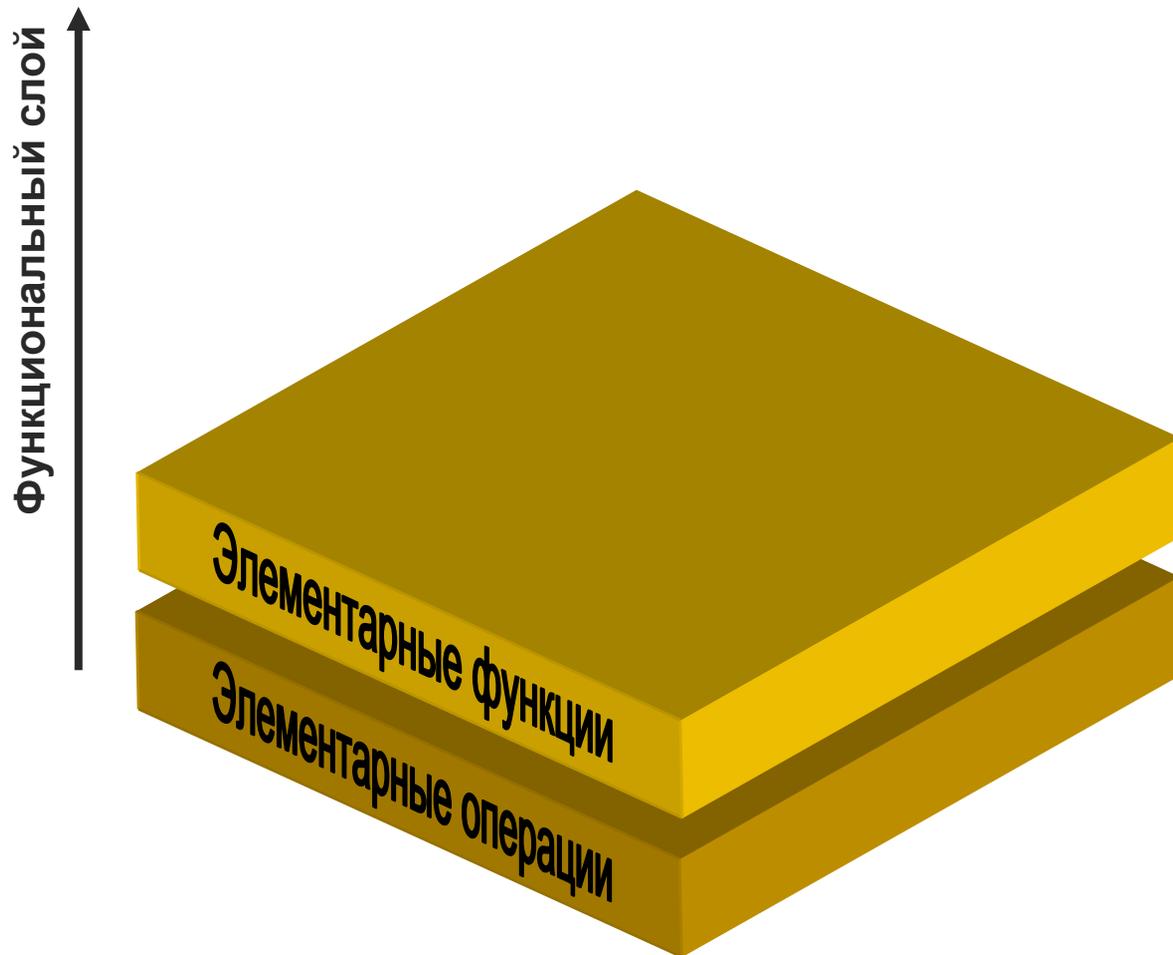
■ Принципы

- Открытость
- Расширенная функциональность
- Обеспечение гибкости в выборе приоритетов вычислений – высокая скорость или высокая точность
 - Выбор базового типа границ интервалов
 - Выбор режима округления
 - Унифицированная спецификация для различных реализаций

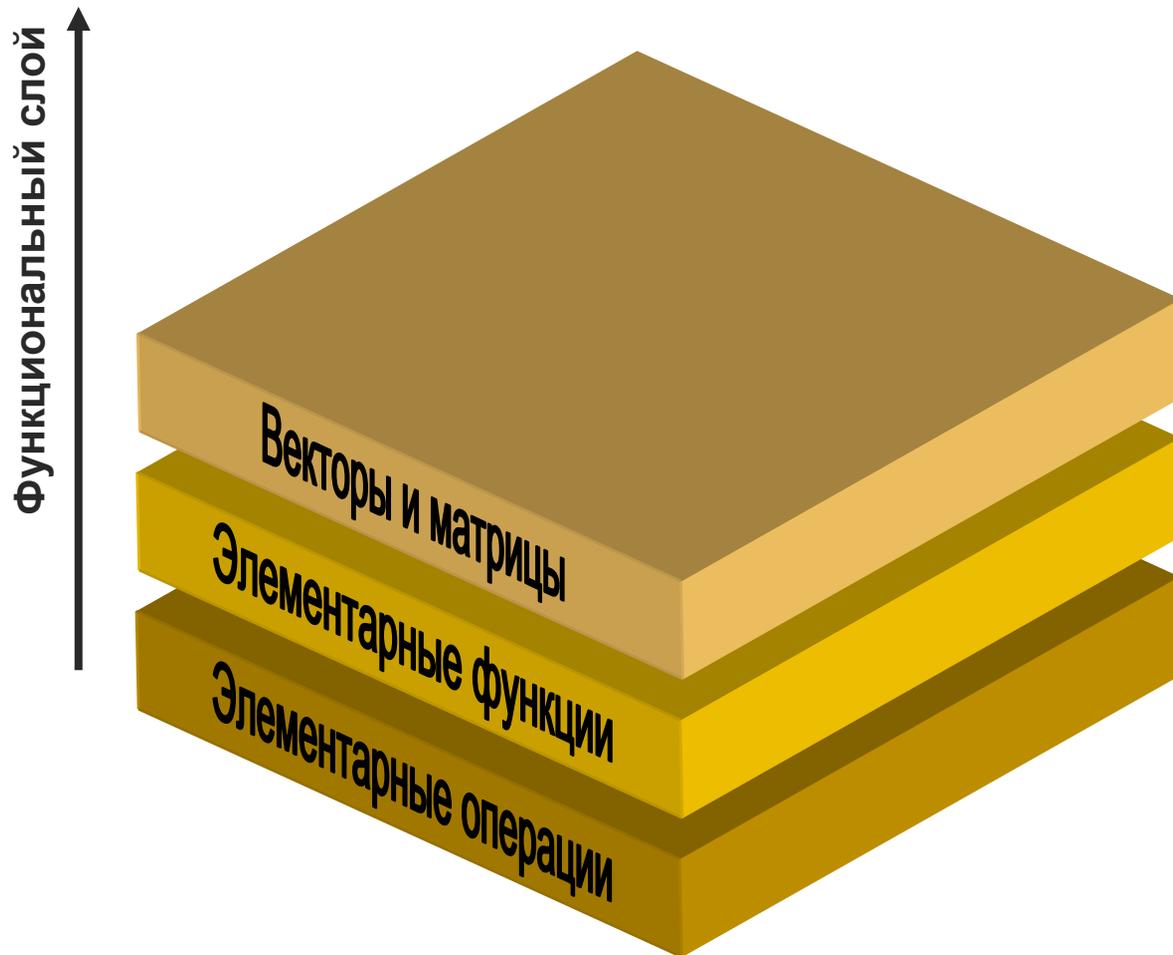
[Структура библиотеки]



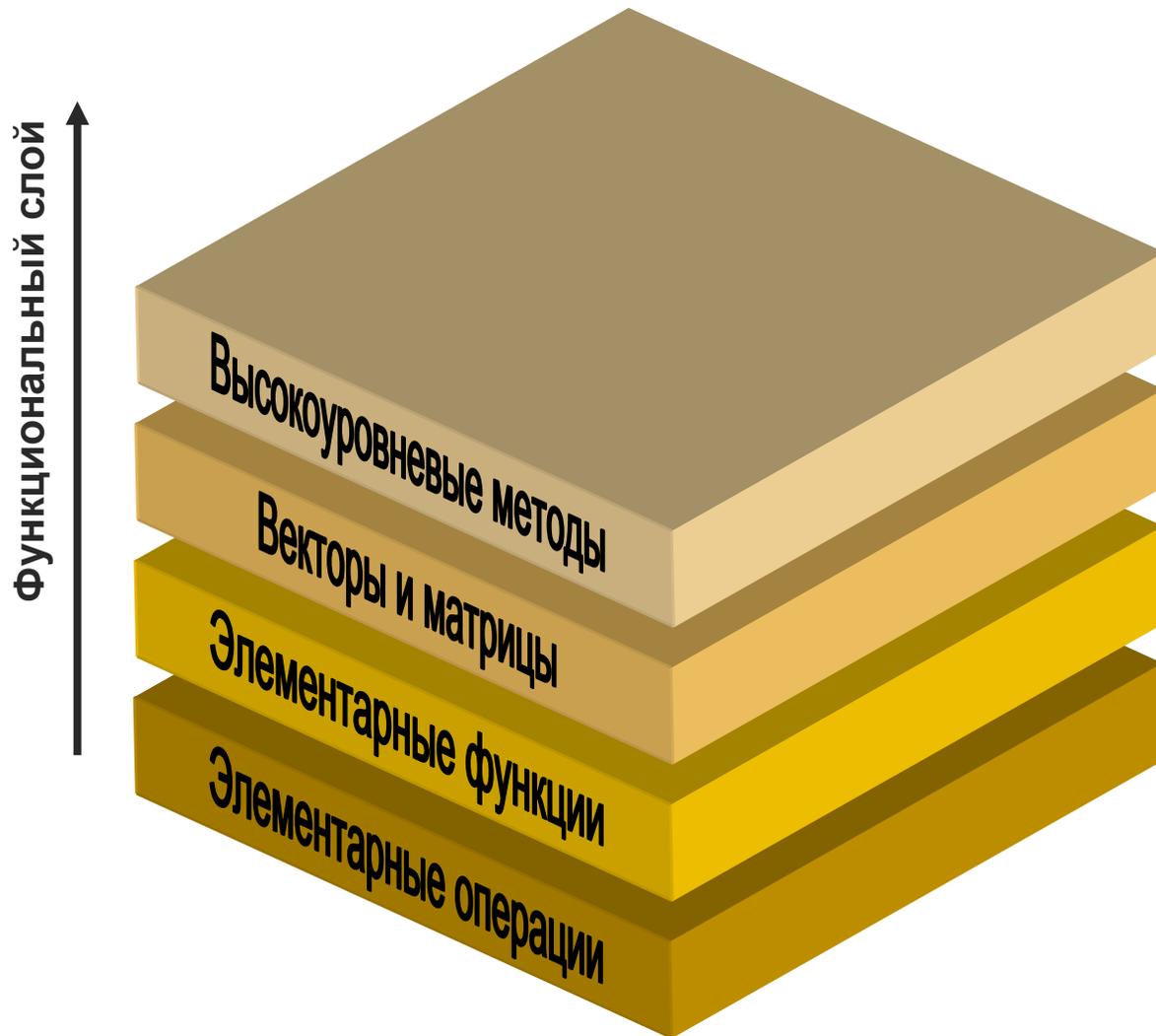
[Структура библиотеки]



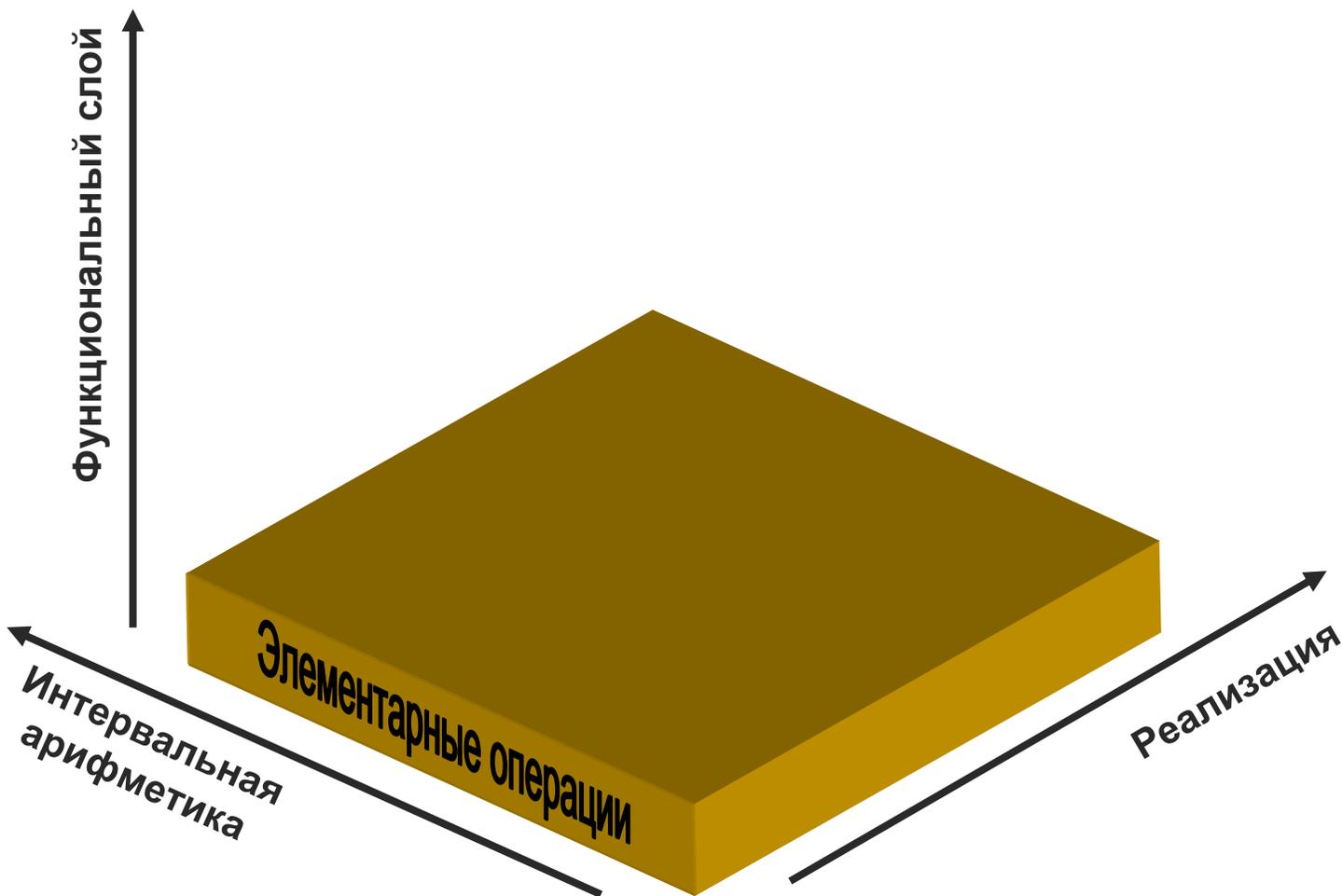
[Структура библиотеки]



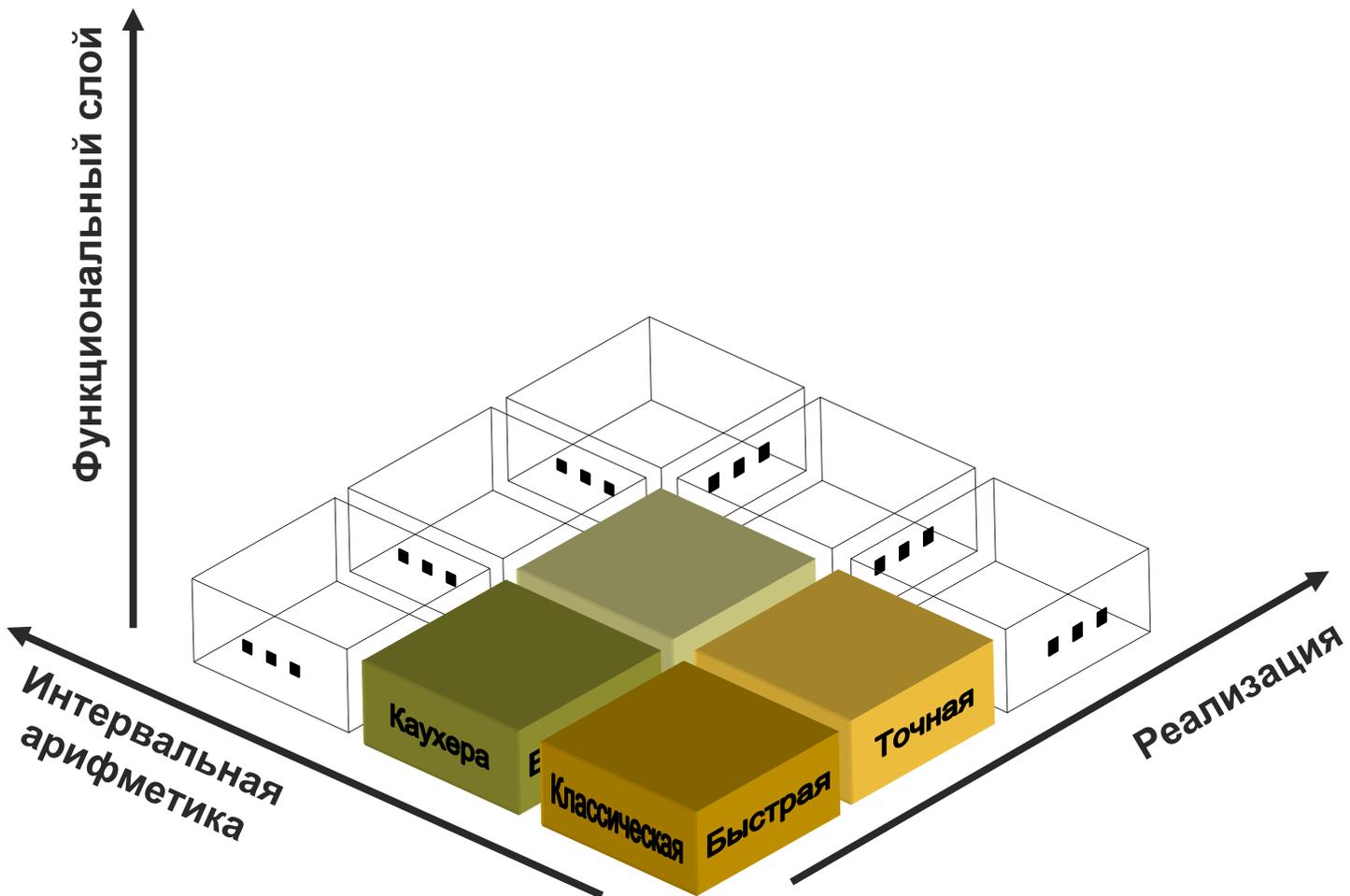
[Структура библиотеки]



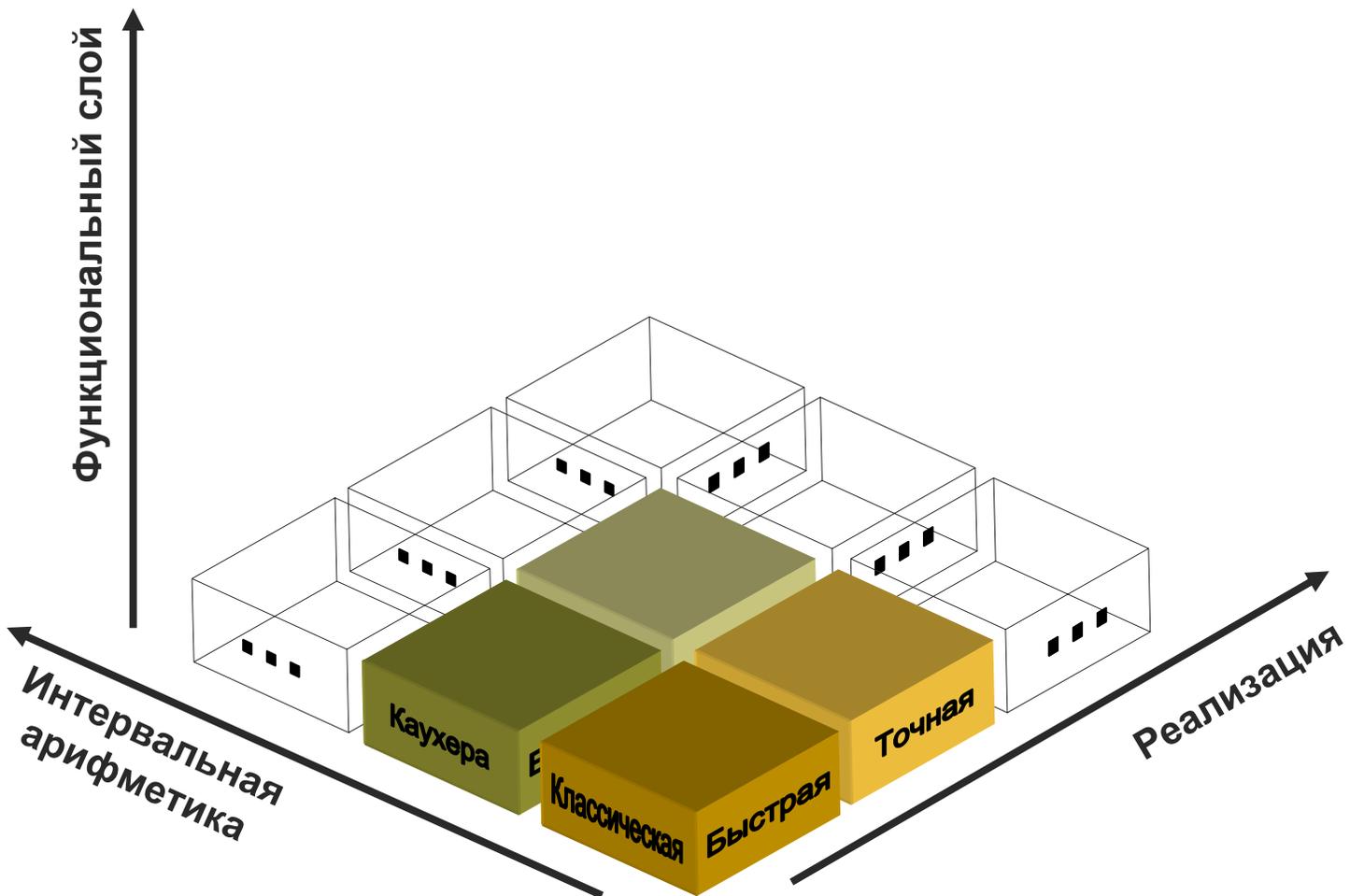
[Структура библиотеки]



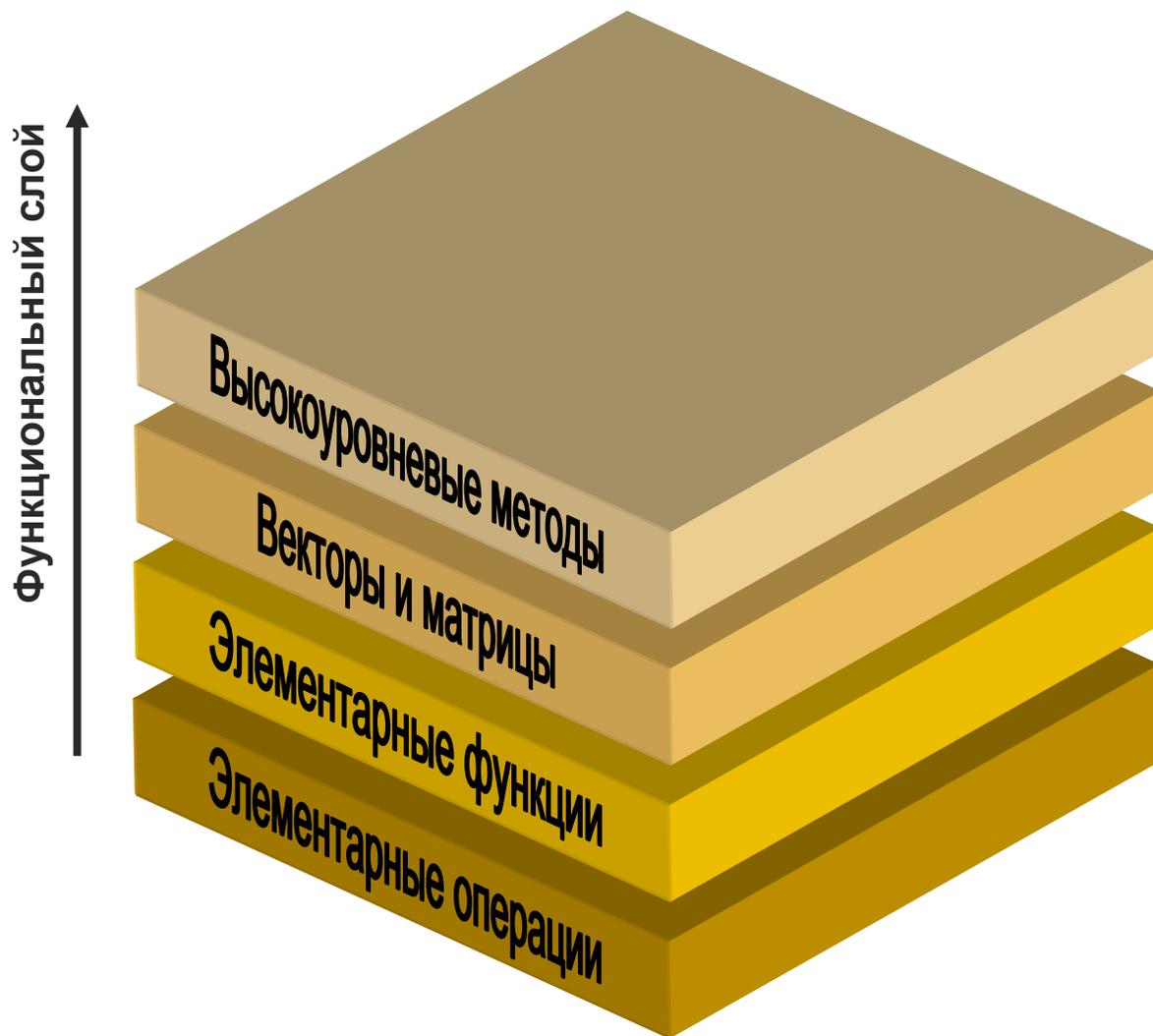
Структура библиотеки



Структура библиотеки



[Структура библиотеки]



Interval с точки зрения пользователя

- «Быстрая» реализация
 - Интервальные типы
 - `DoubleInterval` – классический интервал
 - `KDoubleInterval` – каухеровский интервал
 - Интервальные арифметики
 - `DIO` – классическая арифметика
 - `KDIO` – арифметика Каухера

```
...
DoubleInterval x = new DoubleInterval(1.0,2.0);
DoubleInterval y = new DoubleInterval(0.1,0.2);
DoubleInterval z = DIO.mult(x,y);
System.out.println(z.toString());
...
```

Interval с точки зрения пользователя

- «Быстрая» реализация
 - Интервальные типы
 - `DoubleInterval` – классический интервал
 - `KDoubleInterval` – каухеровский интервал
 - Интервальные арифметики
 - `DIO` – классическая арифметика
 - `KDIO` – арифметика Каухера

```
...
KDoubleInterval x = new KDoubleInterval(4.0, 1.0);
KDoubleInterval y = new KDoubleInterval(0.1, 0.2);
KDoubleInterval z = KDIO.mult(x, y);
System.out.println(KDIO.dual(z).toString());
...
```

Interval с точки зрения пользователя

- «Точная» реализация
 - Интервальные типы
 - `IntervalClassic` – классический интервал
 - `IntervalKauher` – каухеровский интервал
 - Интервальные арифметики
 - `Alg` – арифметика

```
...
Interval x = new IntervalClassic("1.0", "2.0");
Interval y = new IntervalClassic("1.0", "1.0");
Interval z = new IntervalClassic("2.0", "3.0");;
Interval res;
res = Alg.div(Alg.sub(Alg.mul(x, Alg.add(x, y)),
    Alg.sub(Alg.sub(Alg.mul(x, y), z), x)), Alg.mul(z, y));
...
```

Interval с точки зрения пользователя

- «Быстрые» интервальные матрицы
 - Типы интервальных матриц
 - `DIntervalMatrix` – матрица из классических интервалов
 - `KDIntervalMatrix` – матрица из каухеровских интервалов
 - Интервальные матричные операции
 - `DIMO` – на основе классической арифметики
 - `KDIMO` – на основе арифметики Каухера

Interval с точки зрения пользователя

■ «Быстрые» интервальные матрицы

```
double[][] aValues = { {1, 1}, {0, 1} };
double[][] bCenter = { {1, 0}, {-1, 1} };
double[][] bRadius = { {0, 0}, {0, 0} };
DoubleInterval[][] cValues = {
{new DoubleInterval(-1, 1), new DoubleInterval(0)},
{new DoubleInterval(0), new DoubleInterval(-1, 1)} };

DIntervalMatrix A = new DIntervalMatrix( aValues );
DIntervalMatrix B = new DIntervalMatrix( bCenter, bRadius );
DIntervalMatrix C = new DIntervalMatrix( cValues );

DIntervalMatrix D = DIMO.mult( DIMO.mult(A, B), C );
DIntervalMatrix E = DIMO.mult( A, DIMO.mult(B, C) );
```

```
D = [ 0.0, 0.0 ], [ -1, 1 ]      E = [ -2.0, 2.0 ], [ -1.0, 1.0 ]
     [ -1.0, 0.0 ], [ -1, 1 ]      [ -1.0, 1.0 ], [ 1.0, 1.0 ]
```

Interval с точки зрения пользователя

- «Быстрые» высокоуровневые методы
 - DIILM – класс методов решения ИСЛАУ
 - Субдифференциальный метод Ньютона нахождения формального решения ИСЛАУ

```
DIILM.newtonSubdiff(c, v, eps, tau);
```

c – матрица коэффициентов,

v – вектор правой части,

eps – требуемая погрешность,

tau – релаксационный параметр

Interval с точки зрения пользователя

- «Быстрый» субдифференциальный метод Ньютона

```
DoubleInterval[][] cValues = {
    {new DoubleInterval(2, 4), new DoubleInterval(-1, 2)},
    {new DoubleInterval(-2, 1), new DoubleInterval(2, 4)} };
DoubleInterval[][] dValues = {
    {new DoubleInterval(-2, 2)},
    {new DoubleInterval(-2, 2)} };

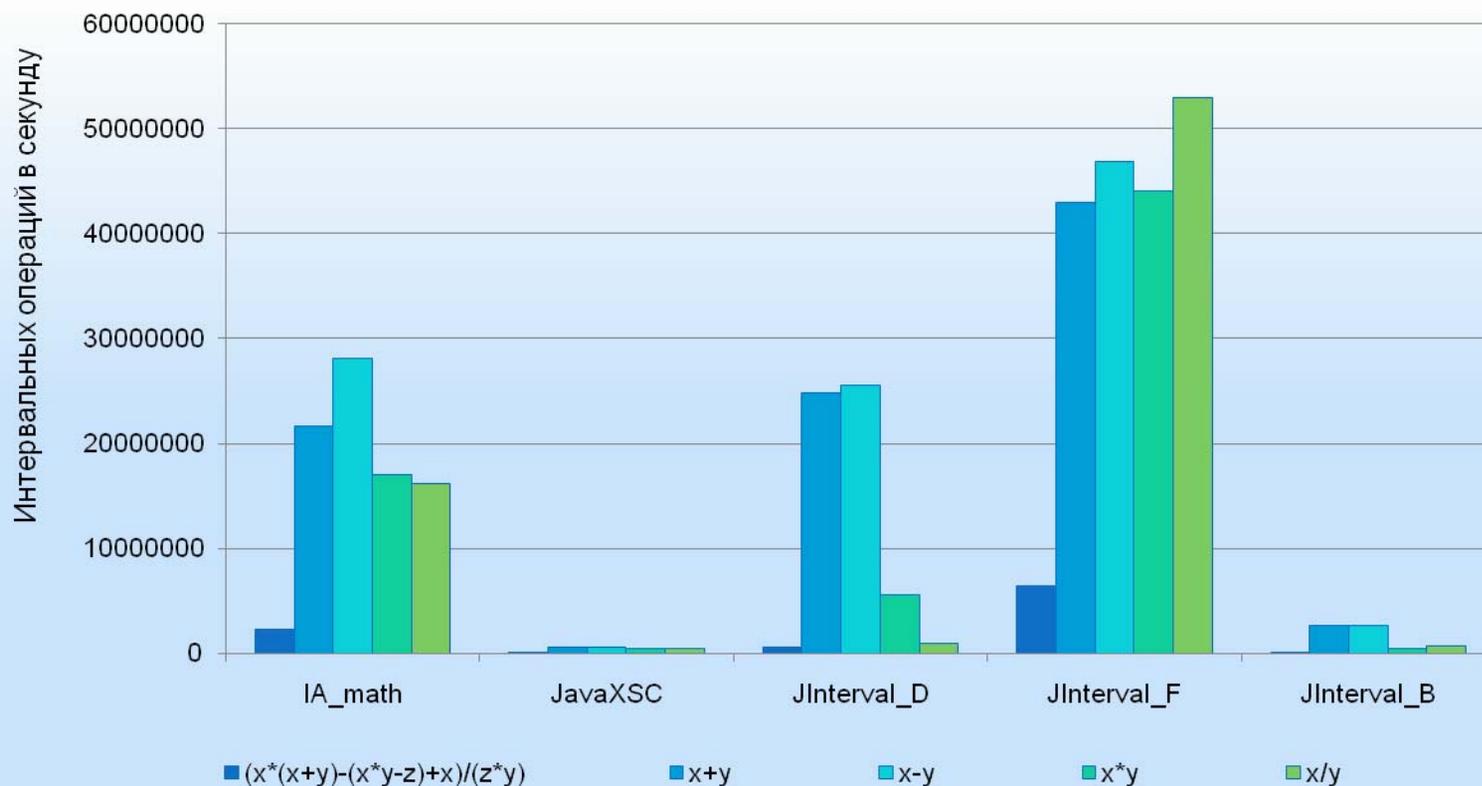
DIntervalMatrix C = new DIntervalMatrix( cValues );
DIntervalMatrix D = new DIntervalMatrix( dValues );

DIntervalMatrix X = DILM.newtonSubdiff( C, D, .0001, 1 );
```

```
X = [ -0.3333333333333333, 0.3333333333333333 ]
     [ -0.3333333333333333, 0.3333333333333333 ]
```

Сравнительные тесты

Производительность



ИТОГИ

- Рабочая версия исходных кодов JInterval:
<http://code.google.com/p/javaintervalmathasu/source/checkout>
- JInterval в текущем состоянии предоставляет «быструю» и «точную» реализации
 - Классической и полной каухеровской ИА
 - Элементарных интервальных функций
 - Интервальных векторов и матриц
 - Метода формального решения ИСЛАУ
- Ближайшие шаги
 - приведение архитектуры в соответствие с изложенными принципами
 - универсализация синтаксиса различных реализаций
 - расширение функционала