

Кумир-2014

Яковлев Виктор Вадимович
НИИСИ РАН

<http://www.niisi.ru/kumir/>
<http://lpm.org.ru/kumir/>
<https://gitorious.org/kumir2>

(официальная страница НИИСИ РАН)
(страница разрабатываемой версии)
(исходные тексты под лицензией GNU GPL2)

План доклада

Часть I. Обзор Кумир 2.1 (бета-версия)

Часть II. Генерация кода с помощью LLVM

<http://www.niisi.ru/kumir/>

<http://lpm.org.ru/kumir/>

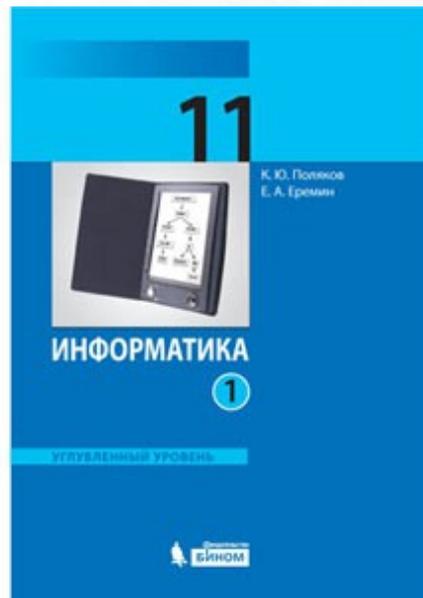
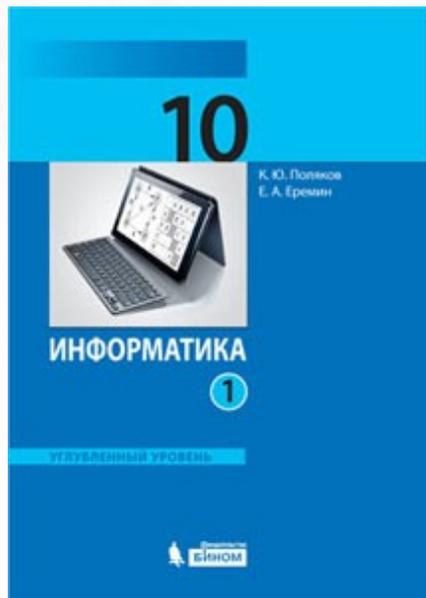
<https://gitorious.org/kumir2>

(официальная страница НИИСИ РАН)

(страница разрабатываемой версии)

(исходные тексты под лицензией GNU GPL2)

Методическая поддержка



Учебник включен в
Федеральный перечень
учебников,
рекомендованных к
использованию на
2013-2014 учебный год

Информатика. Углубленный уровень : учеб. для 10 кл.: в 2 ч. /
К. Ю. Поляков, Е. А. Еремин. - М.: БИНОМ. Лаб. знаний, 2013

Информатика. Углубленный уровень : учеб. для 11 кл.: в 2 ч. /
К. Ю. Поляков, Е. А. Еремин. - М.: БИНОМ. Лаб. знаний, 2013

Методическая поддержка



Учебники Кушниренко
`bin/kumir2-classic`



Учебник Полякова
`bin/kumir2-highgrade`

Конфигурации системы Кумир



Классический Кумир

Соответствует
функциональности
Кумир 1.x



Кумир с вкладками

- + возможность открытия нескольких файлов
- + встроенный текстовый редактор для файлов данных
- + стартовая страница

Кумир для старших классов

+ исполнители «Строки» и «Файлы» являются частью языка (в соответствии с учебником Полякова)



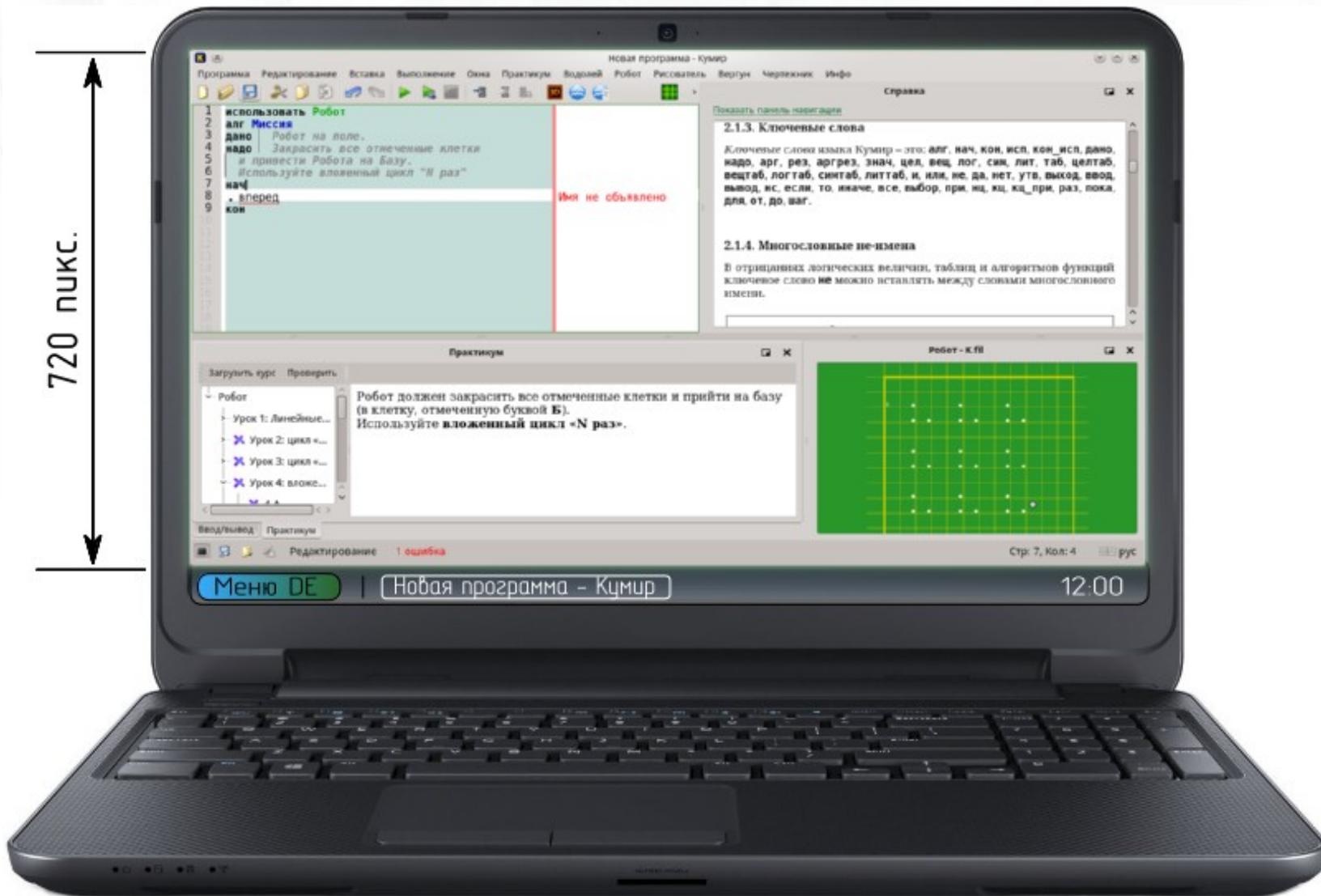
Кумир для учителей

+ возможность редактирования дополнительных атрибутов заданий (скрытые и защищенные строки, создание специальных алгоритмов)

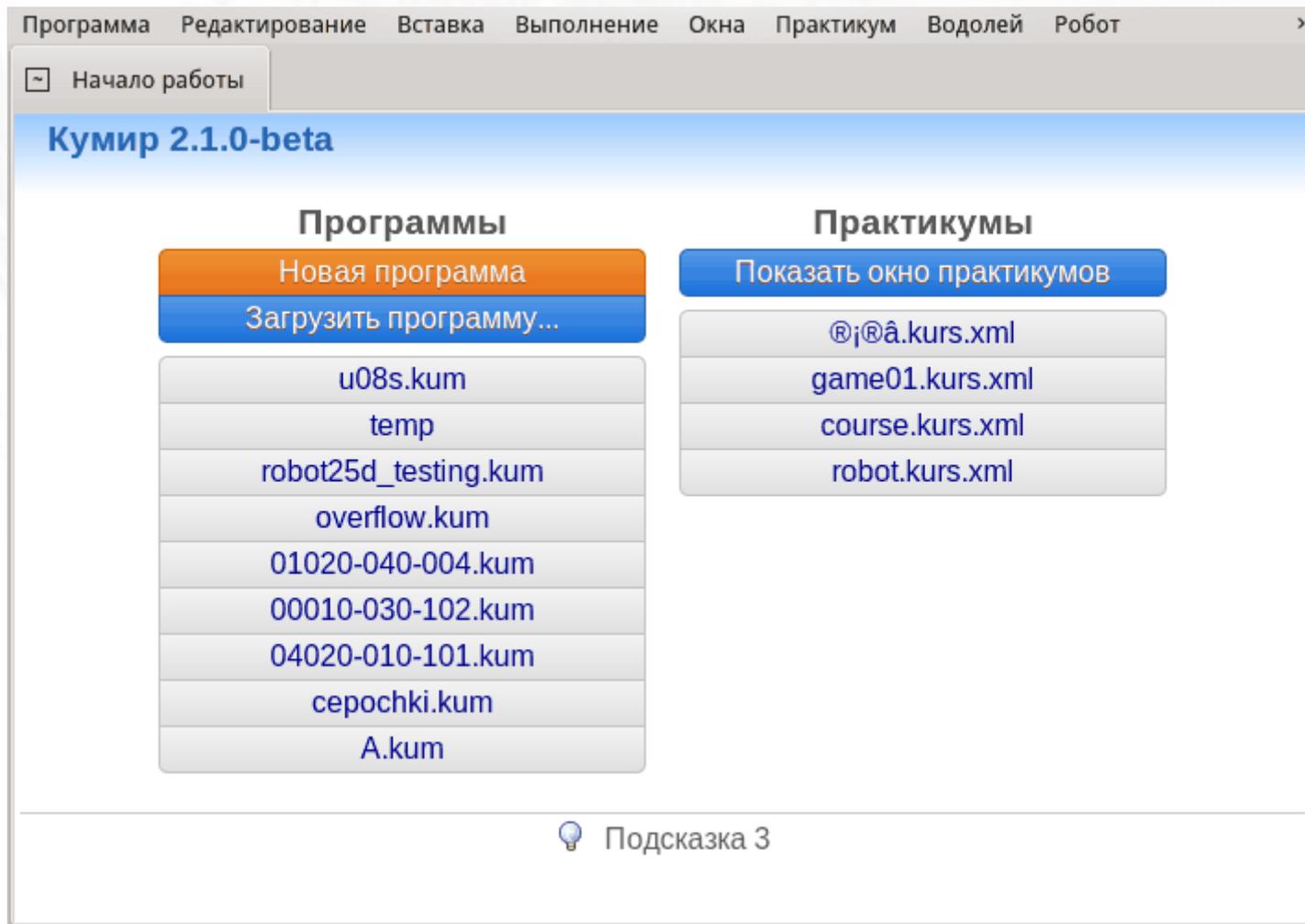
Оконный интерфейс системы Кумир

1280 пикс.

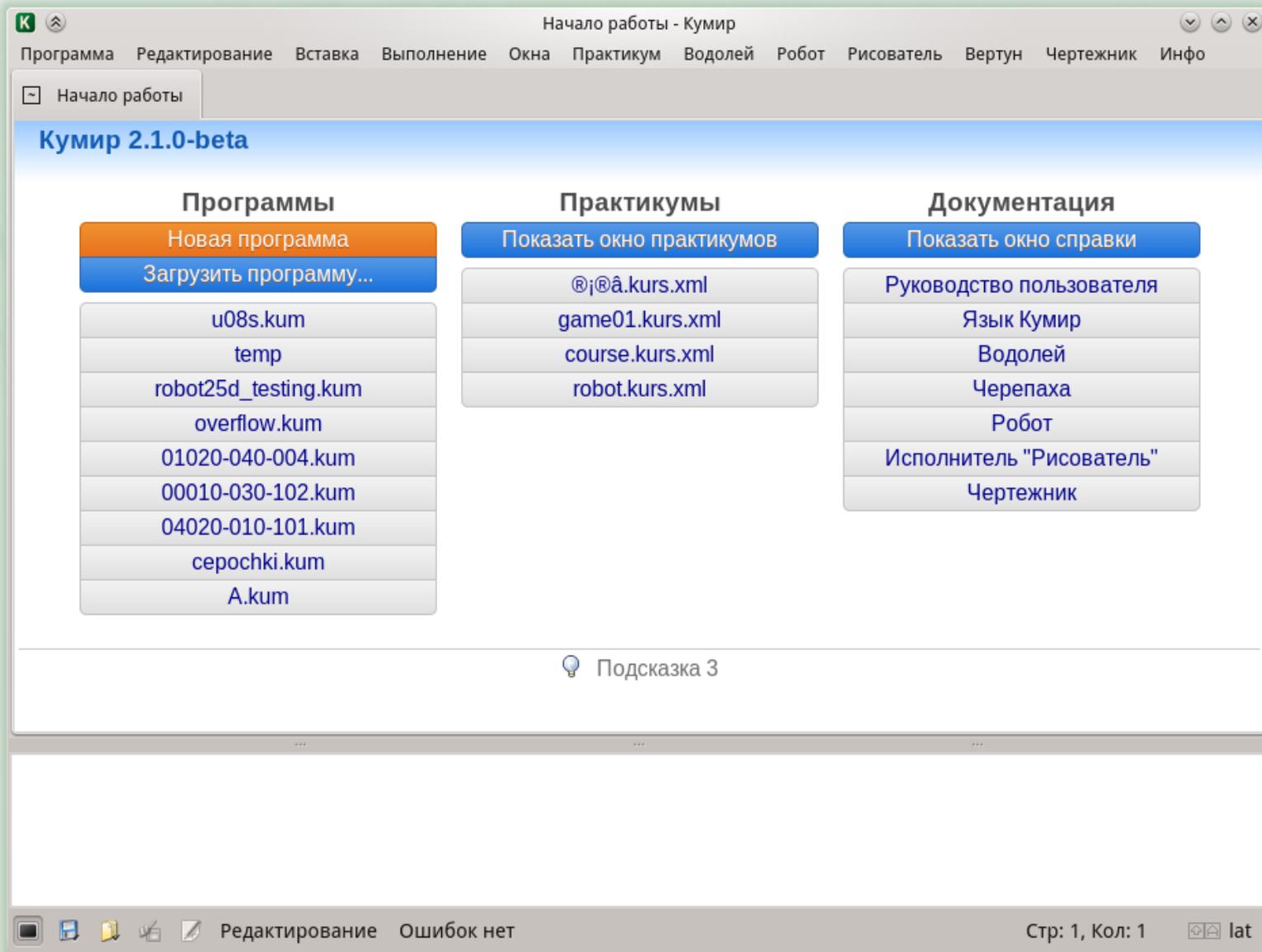
720 пикс.



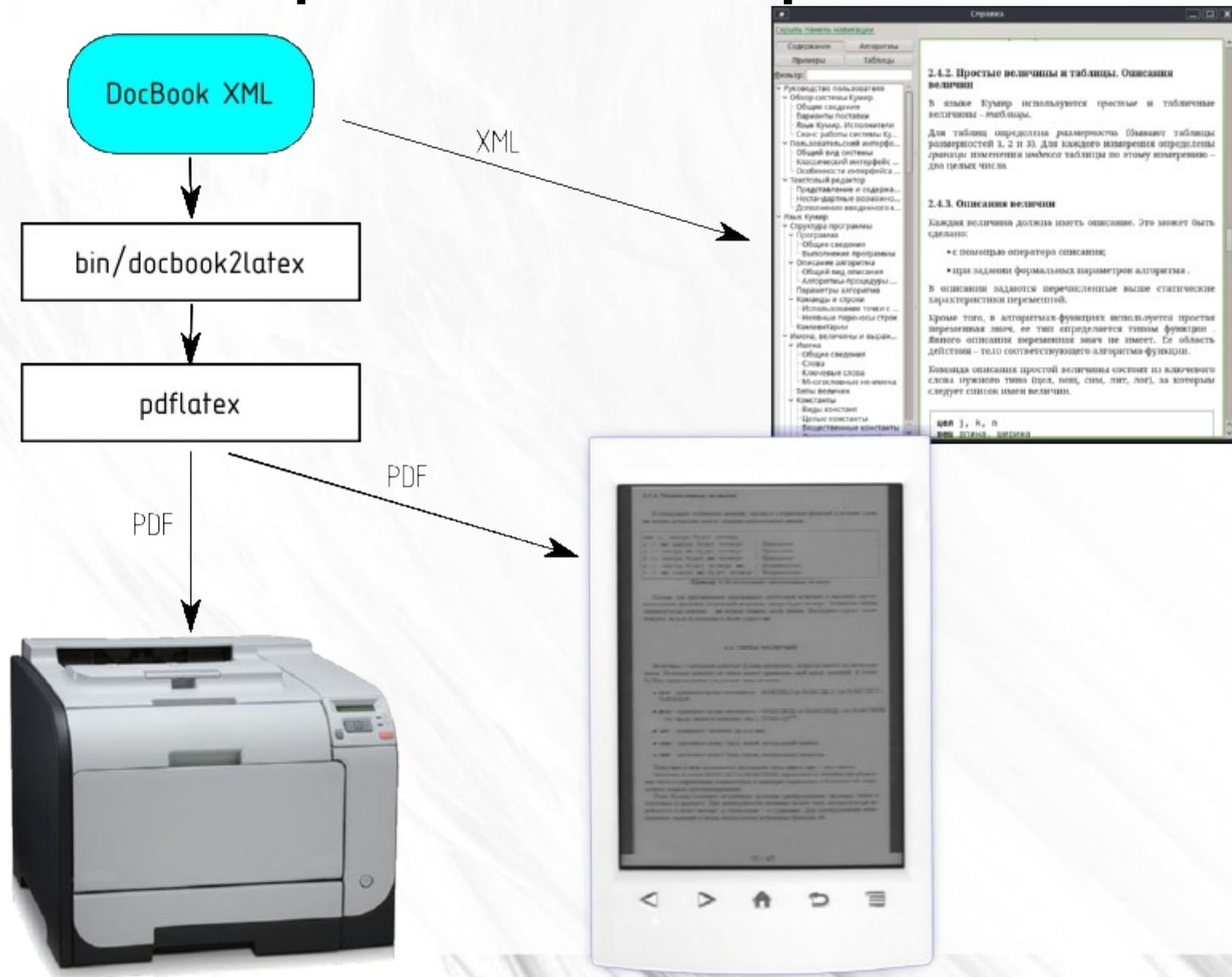
Вариант интерфейса с вкладками `bin/kumir2-ide`



Стартовая страница bin/kumir2-ide

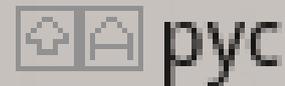


Встроенная справка



Редактор кода и раскладки клавиатуры

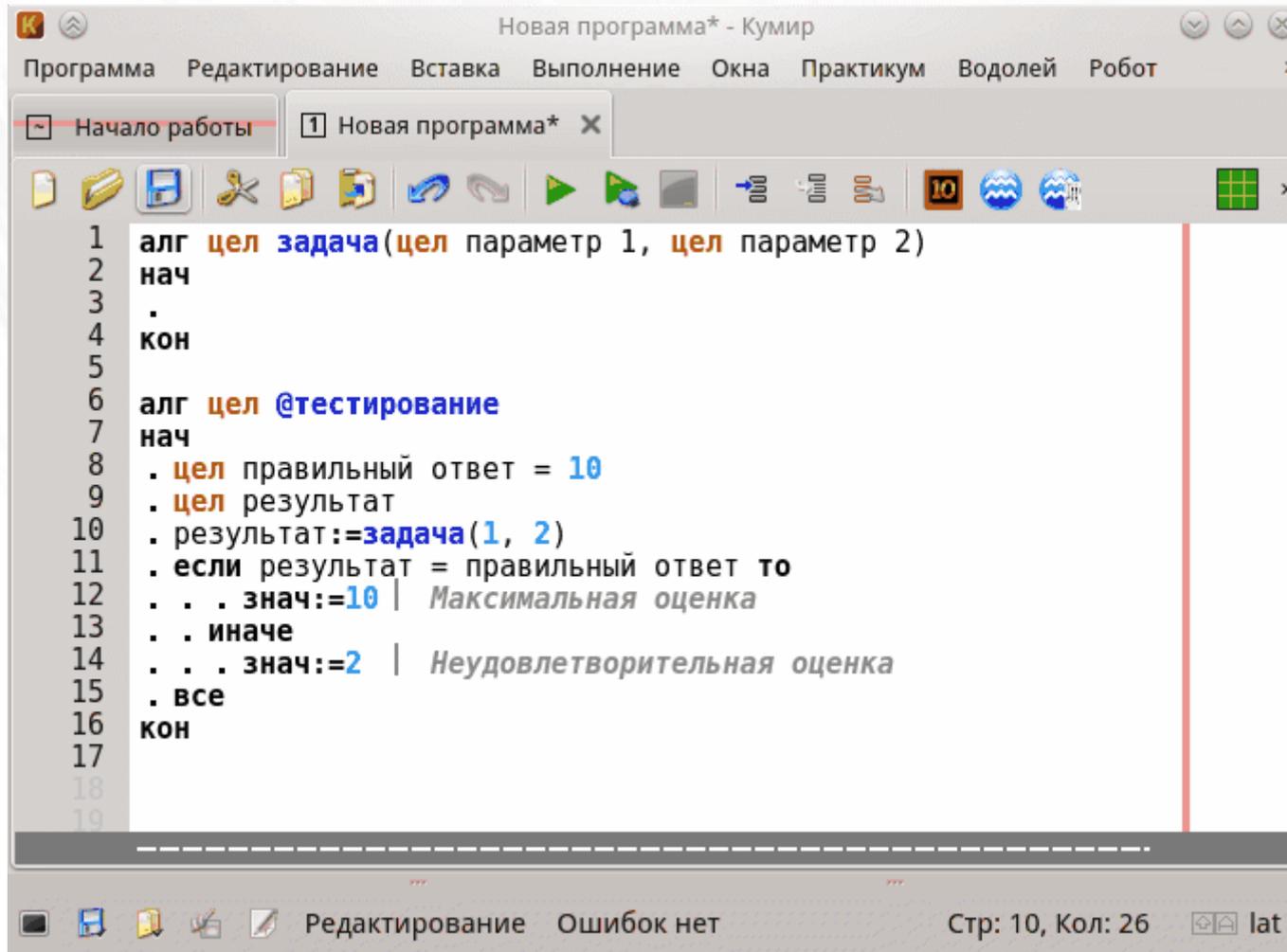
- Нажатие клавиши **ALT** во время набора текста временно переключает раскладку
- Состояние клавиш-модификаторов отображается в строке статуса



Редактор кода: семантическое автодополнение текста

- Возможно автодополнение текста по нажатию **TAB** или **CTRL+ПРОБЕЛ**
- Список предлагаемых вариантов вставки зависит от положения курсора в тексте программы
- Для уменьшения количества синтаксических ошибок, завершающие ключевые слова составных конструкций дописываются автоматически (опция в настройках)

Редактирование программ учителем



Новая программа* - Кумир

Программа Редактирование Вставка Выполнение Окна Практикум Водолей Робот

Начало работы 1 Новая программа* X

```
1 алг цел задача(цел параметр 1, цел параметр 2)
2 нач
3 .
4 кон
5
6 алг цел @тестирование
7 нач
8 . цел правильный ответ = 10
9 . цел результат
10 . результат:=задача(1, 2)
11 . если результат = правильный ответ то
12 . . . знач:=10 | Максимальная оценка
13 . . иначе
14 . . . знач:=2 | Неудовлетворительная оценка
15 . все
16 кон
17
18
19
```

Редактирование Ошибок нет Стр: 10, Кол: 26 lat

Непрерывная интеграция

<http://teamcity.lpm.org.ru/>
<http://youtrack.lpm.org.ru/>

- Build Service
- Bug Tracking

Цель сборки	Платформа	Результат
Тестирование регрессий	Linux	Файл отчета (.txt) + сообщение в BTS
Забота о пользователях Windows	Win32	Файл установщика NSIS (.exe)
Забота о фанатиках Apple	MacOS	Ничего (отсутствие ошибок сборки)

Конец первой части

Недоработки версии 2.1, блокирующие выпуск релиза (релиз-кандидата)

1. Тексты документации
2. Artwork (в первую очередь – иконки)
3. Тестирование на реальных пользователях

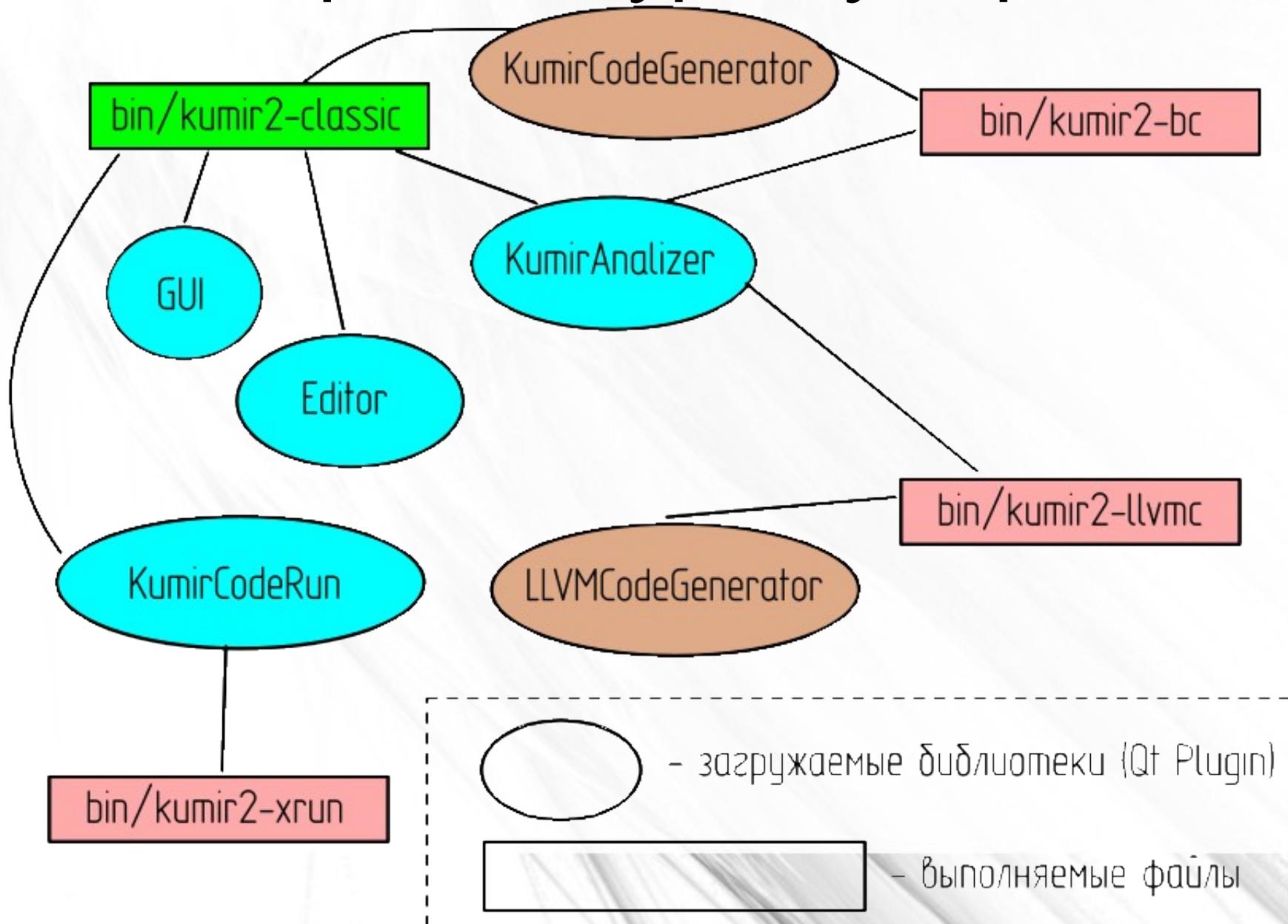
Часть II. Генератор кода LLVM

```
int main()
{
    float result = 0;
    printf("Hello, World\n");
    return result;
}
```

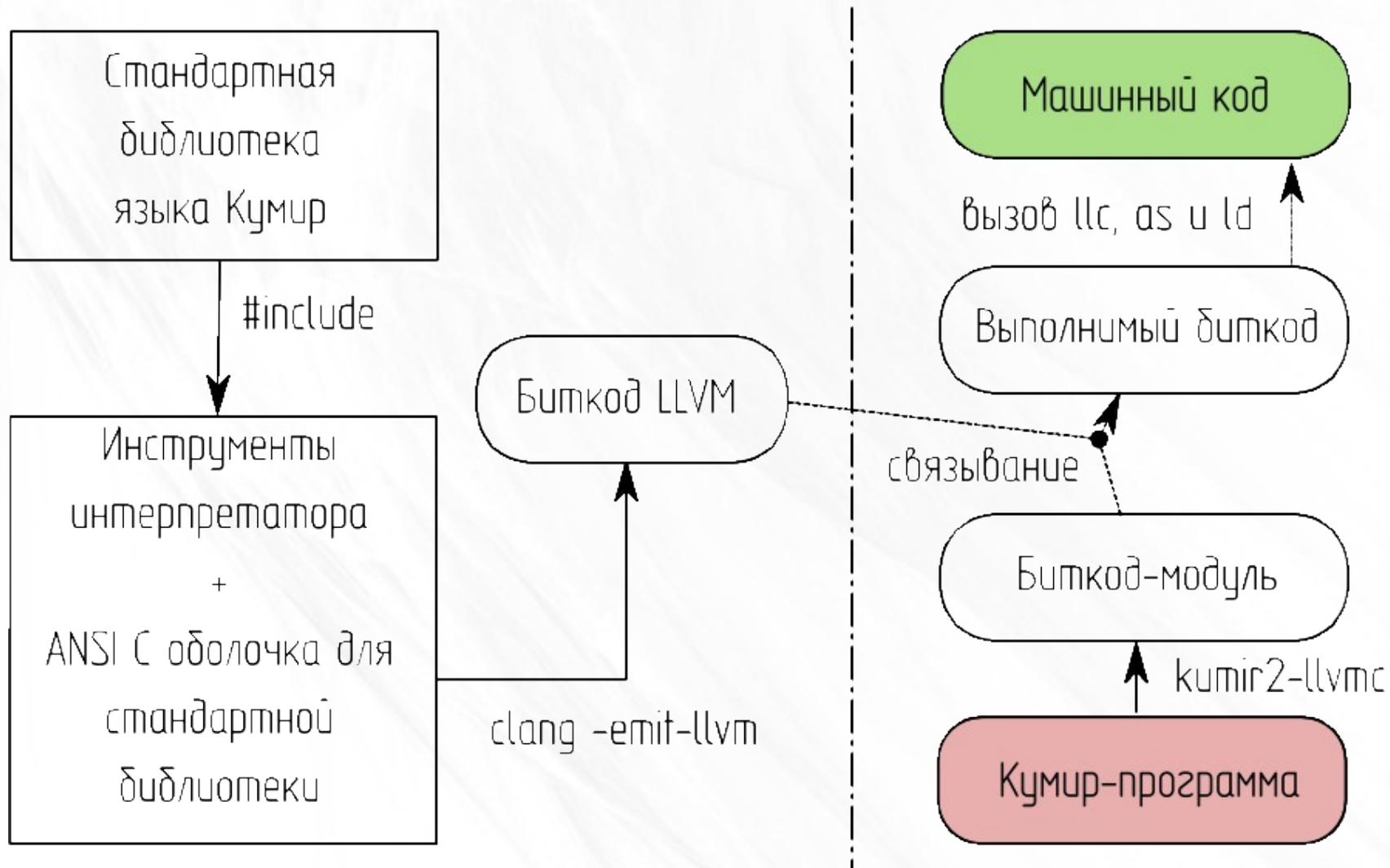
```
> # компиляция C-программы в биткод
> clang -c -O0 -emit-llvm -o hello.bc hello.c
> # декодирование биткода в текст
> llvm-dis hello.bc
> # запуск биткода в JIT-интерпретаторе (x86)
> lli hello.ll
Hello, World
```

```
@.str = private unnamed_addr constant [14 x i8] c"Hello, World\0A\00", align 1
define i32 @main() {
    %1 = alloca i32, align 4
    %result = alloca float, align 4 ; выделение памяти в стеке для result
    store i32 0, i32* %1
    store float 0.000000e+00, float* %result, align 4 ; присваивание result = 0
    %2 = call i32 @printf(i8* getelementptr inbounds
                        ([14 x i8]* @.str, i32 0, i32 0))
                        ; вызов printf с аргументом - указателем на 0 элемент строки
    %3 = load float* %result, align 4 ; загрузка значения result в регистр
    %4 = fptosi float %3 to i32 ; явное приведение типа float -> signed int
    ret i32 %4 ; выход из функции с возвратом значения
}
declare i32 @printf(i8*, ...)
```

Архитектура Кумир



Реализация LLVM в Кумир



Реализация LLVM в Кумир

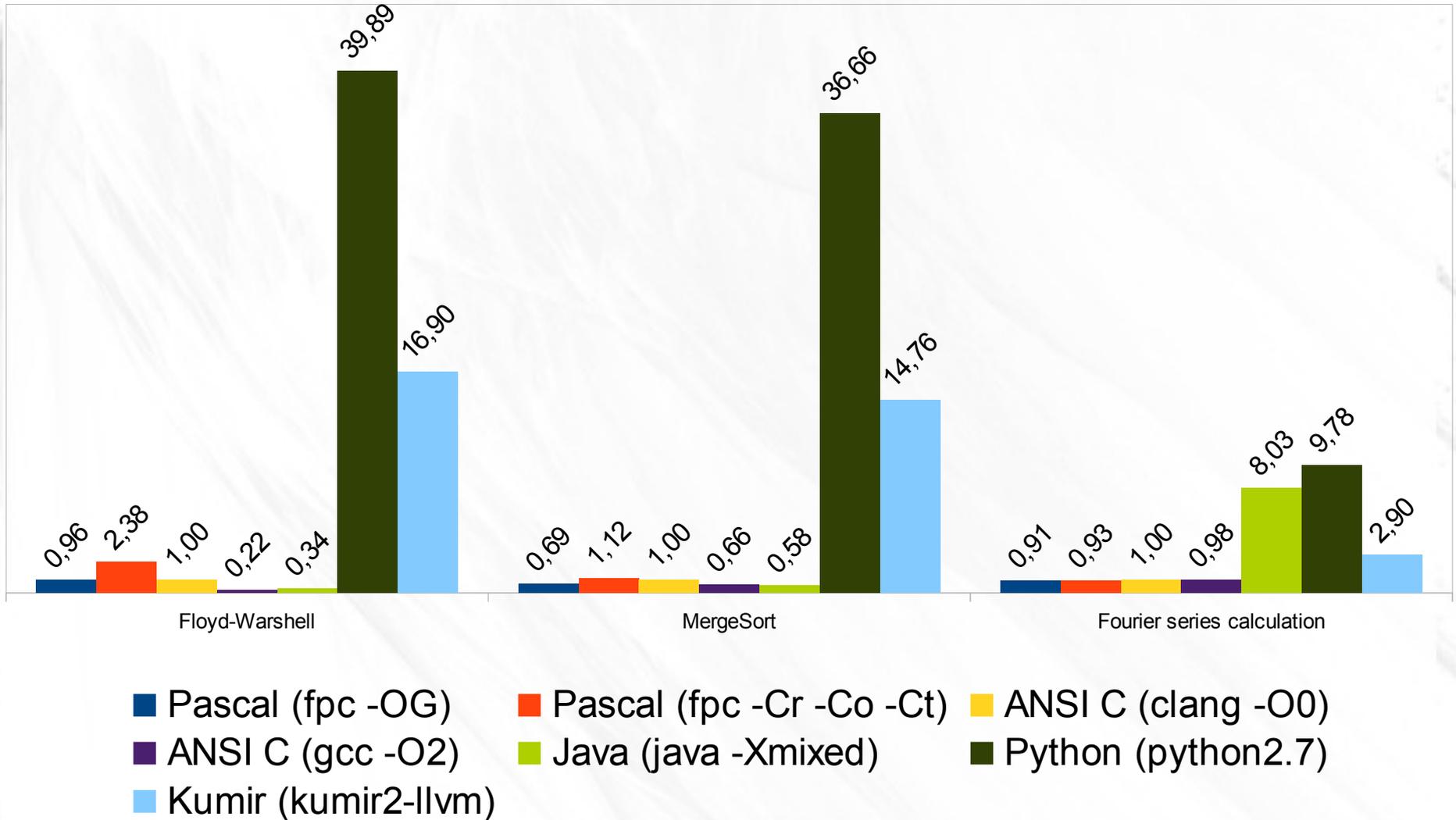
Преимущества

- возможность разработки используя инструменты для C++
- полная совместимость с интерпретатором

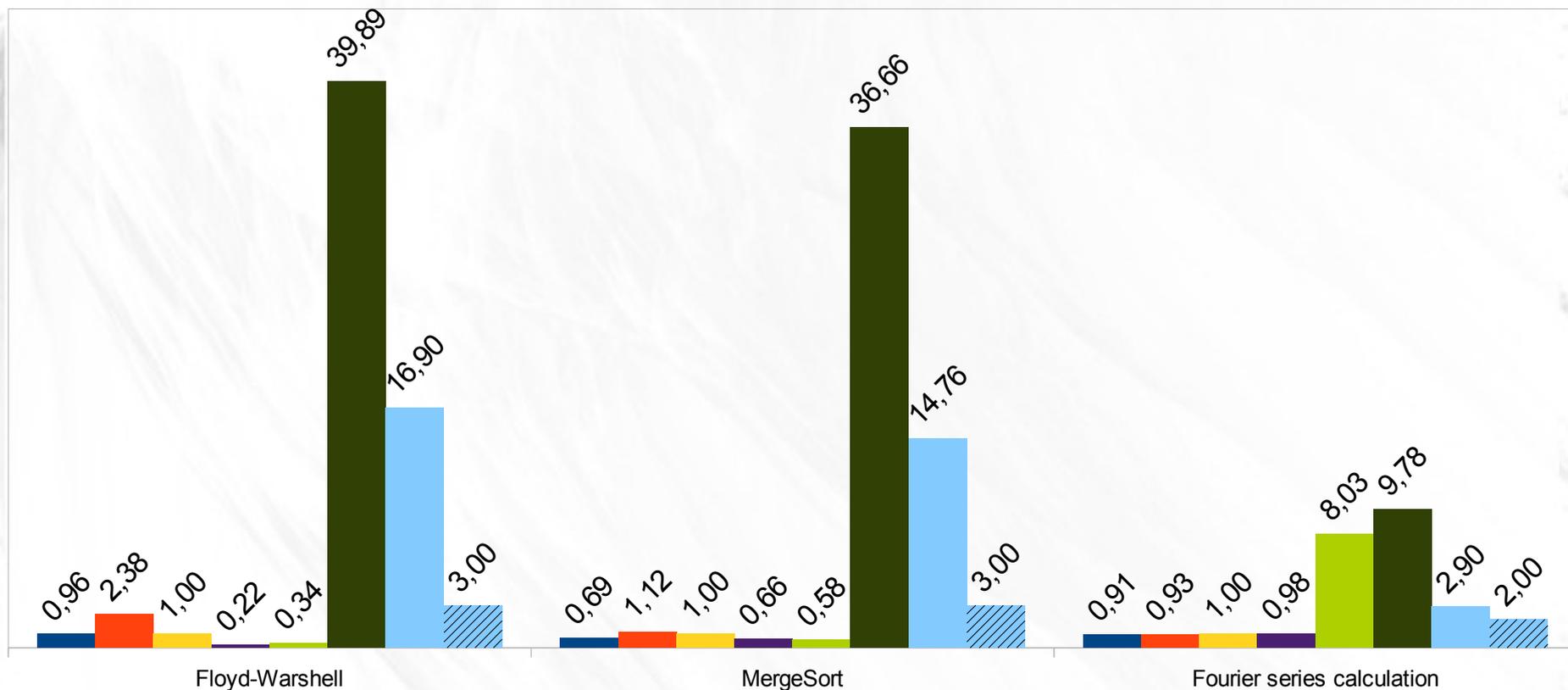
Недостатки

- вызов функций через стек элементарных операций
- использование внутренних типов данных, ориентированных на интерпретатор, неэффективно оперирует регистрами

Сравнение производительности



Пределы роста производительности



■ Pасcal (fpc -OG)

■ Pасcal (fpc -Cr -Co -Ct)

■ ANSI C (clang -O0)

■ ANSI C (gcc -O2)

■ Java (java -Xmixed)

■ Python (python2.7)

■ Кумир (kumir2-llvm)

■ Кумир (в теории)

Результаты

1. Реализован компилятор с языка программирования Кумир в машинный код. Этот компилятор уже можно применять при проведении олимпиад.
2. Существуют резервы для увеличения производительности. Дальнейшее развитие этого направления будет в сторону оптимизации.
3. Версия для Windows зависит от готовности инструментов сторонних разработчиков.

Кумир-2014
Яковлев Виктор Вадимович
НИИСИ РАН

Спасибо за внимание!
Вопросы?

<http://www.niisi.ru/kumir/>
<http://lpm.org.ru/kumir/>
<https://gitorious.org/kumir2>

(официальная страница НИИСИ РАН)
(страница разрабатываемой версии)
(исходные тексты под лицензией GNU GPL2)