

Разработка интерпретатора языка программирования Emacs Lisp Эмуляция команд текстового редактора GNU Emacs для платформы IntelliJ IDEA

Студент: Полищук Екатерина

Руководитель: Тузова Екатерина



Введение

- Текстовый редактор GNU Emacs
 - Множество расширений
 - Широкие возможности для индивидуальной настройки
- IDE IntelliJ IDEA
 - Обширный функционал в помощь разработчику

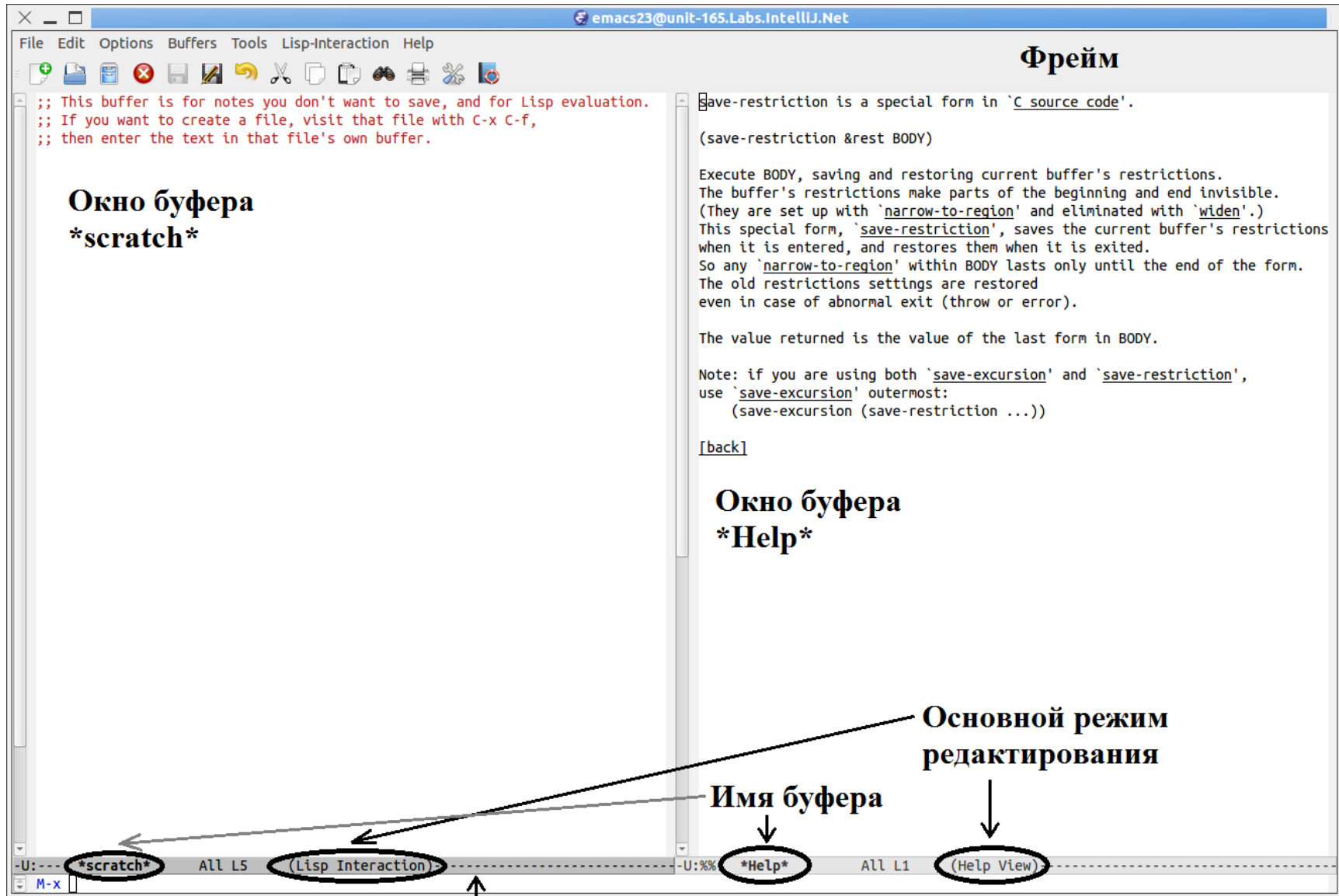
Постановка задачи

- Поддержка функционала GNU Emacs в IDE IntelliJ IDEA
- Расширение (= плагин) для IntelliJ IDEA
- Требования:
 - Аналогичное редактирование текста
 - Те же понятия графических интерфейсов
 - Key bindings
 - Поддержка языка программирования Emacs Lisp
 - Поддержка расширений для GNU Emacs

Предметная область

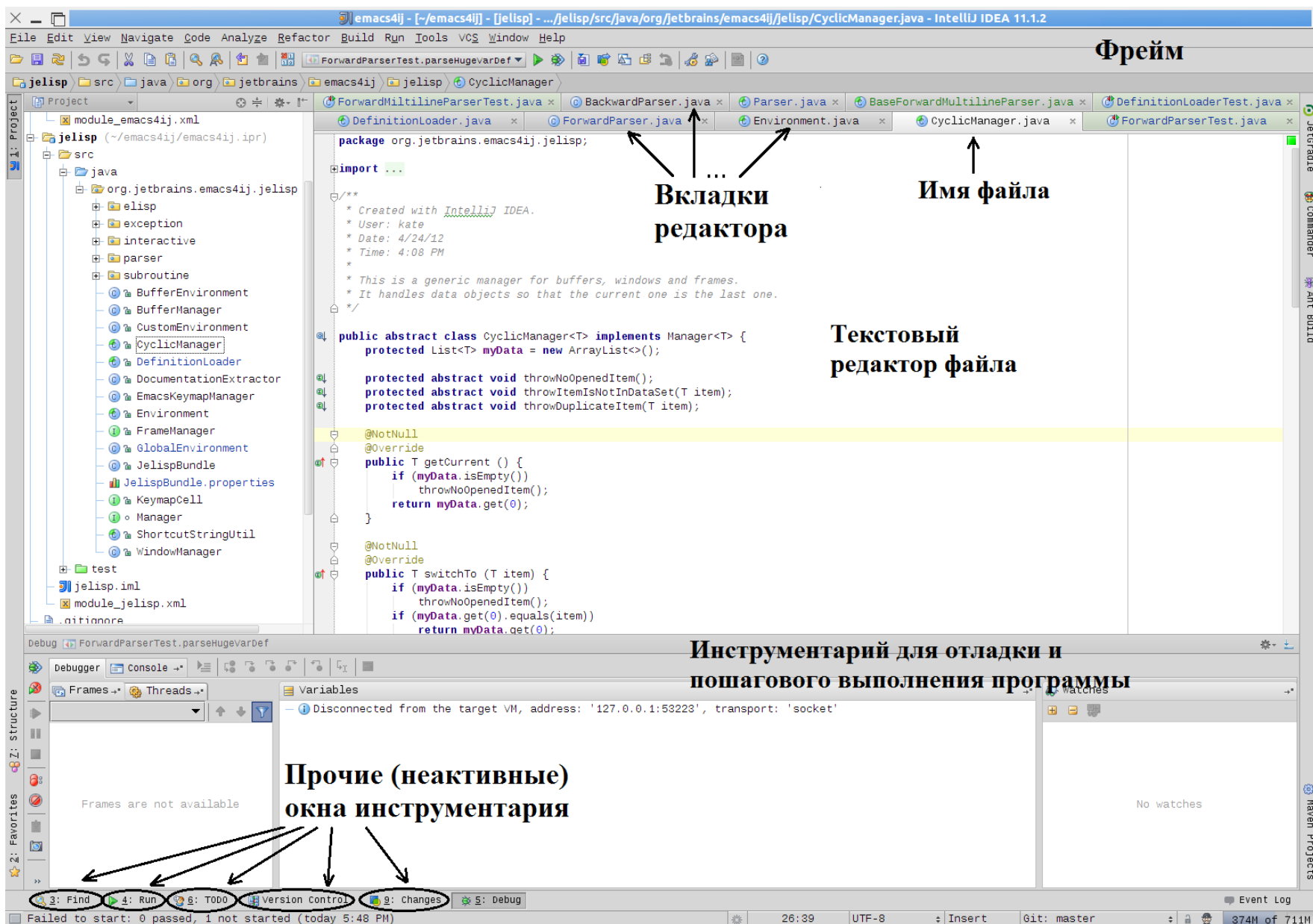
- GNU Emacs =
= интерпретатор Emacs Lisp на C +
+ исходный код на Emacs Lisp +
+ [расширения на Emacs Lisp]
- Emacs Lisp (Dynamic scoping) \neq Common Lisp (lexical scoping)

Интерфейс GNU Emacs



Минибуфер

Интерфейс IntelliJ IDEA



Интерфейс разработанного плагина

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help **Фрейм**

Project: t1 [~/IdeaProjects/t1] | test.lisp x | t2.lisp x

Project Structure: t1 (~/IdeaProjects/t1)

- .idea
- src
 - t2.lisp
 - test.lisp
- t1.iml

 External Libraries

```

M-x
(switch-to-buffer "t2.lisp")
(current-buffer)

(completing-read "buf: " obarray nil nil "ini")
(point)

(defun f (a) (+ a 5)
  (+ 6 2))

(f 5)

(eval-last-sexp nil)
(local-set-key)
(current-global-map)

(string-match " (ick)" "The quick fox jumped quickly.")

(string-match "\\(qu\\)\\(ick\\)"
  "The quick fox jumped quickly.")

;0123456789
- 4                dg

(match-string 0 "The quick fox jumped quickly.")
- "quick"
(match-string 1 "The quick fox jumped quickly.")
- "qu"
(match-string 2 "The quick fox jumped quickly.")
- "ick"

(key-binding "<TAB>")      hjh          (current-buffer)
(following-char)
(emacs-lisp-mode)
major-mode
(font-lock-mode)

(current-local-map)
(current-global-map)
  
```

Минибуфер

Буфер test.lisp

Emacs4ij Echo Area

(#<buffer test.lisp> #<buffer t2.lisp> #<buffer test.lisp> #<buffer t2.lisp> #<buffer *Help*> #<buffer *Minibuf-0*> #<buffer test.lisp>)

Область вывода результатов вычислений

Окно справки

6: TODO | Emacs4ij Echo Area | Emacs4ij Help | Event Log

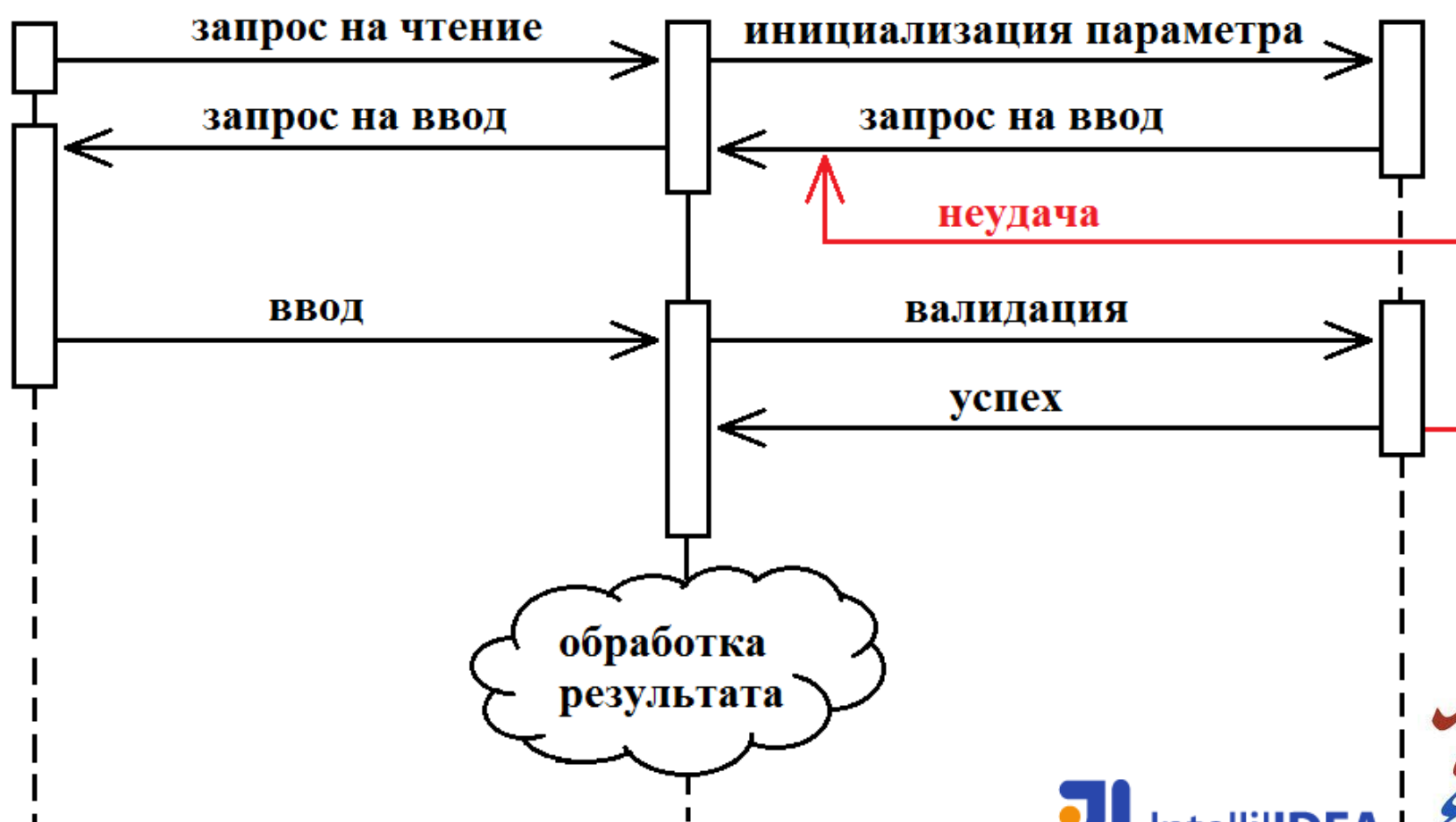
18:17 | UTF-8 | Insert | 374M of 711M

Интерактивное чтение параметра

Пользователь

Минибуфер

Дескриптор
интерактивной формы



Результат: функциональность разработанного плагина

- ✓ Аналогичное редактирование текста
 - ✓ Вставка, удаление, навигация
 - ✓ Режимы редактирования для текста на естественном языке и для кода на Emacs Lisp
 - ✓ Регулярные выражения
- ✓ Те же понятия графических интерфейсов
 - ✓ Минибуфер
 - ✓ Интерактивное выполнение команд
- ✓ Key bindings

Результат 2

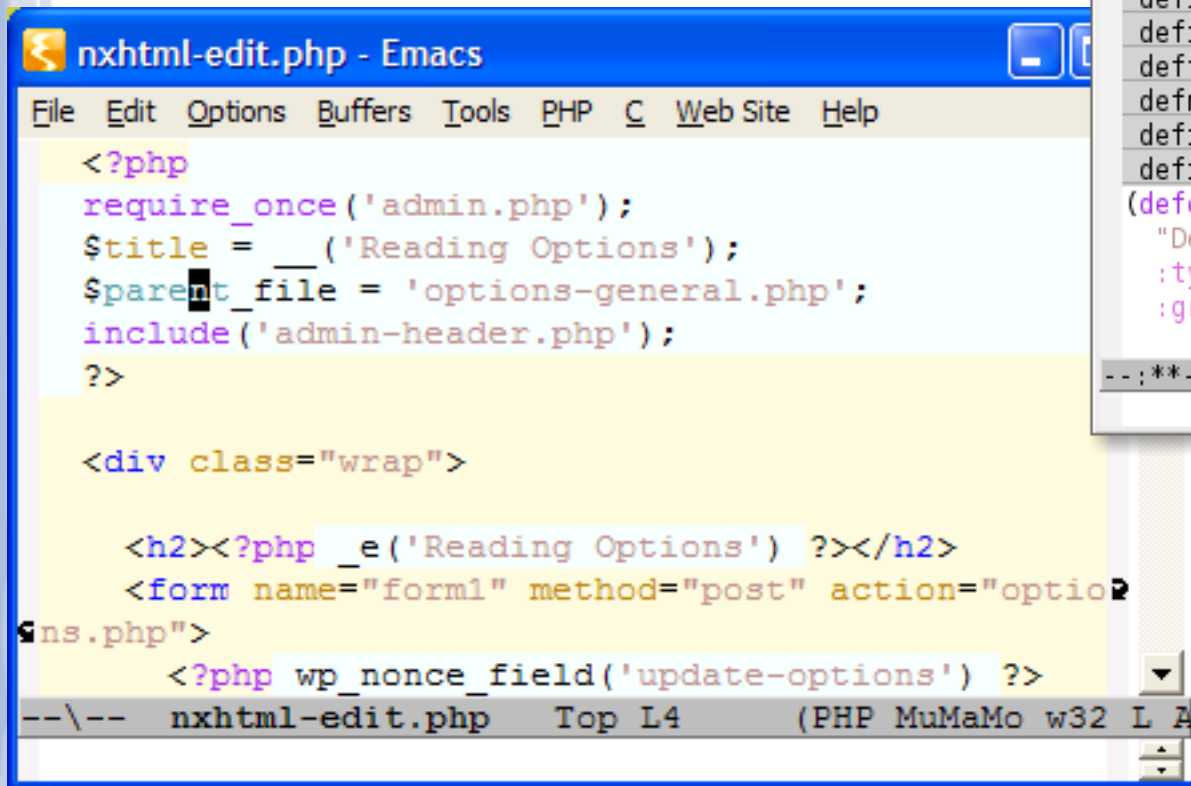
- ✓ Поддержка языка программирования Emacs Lisp
 - ✓ Синтаксический анализатор языка Emacs Lisp
 - ✓ Набор встроенных функций
 - ✓ Индексация исходного кода GNU Emacs
 - ✓ Документация
- ✓ Частичная поддержка расширений для GNU Emacs

Спасибо за внимание!



GNU Emacs modes

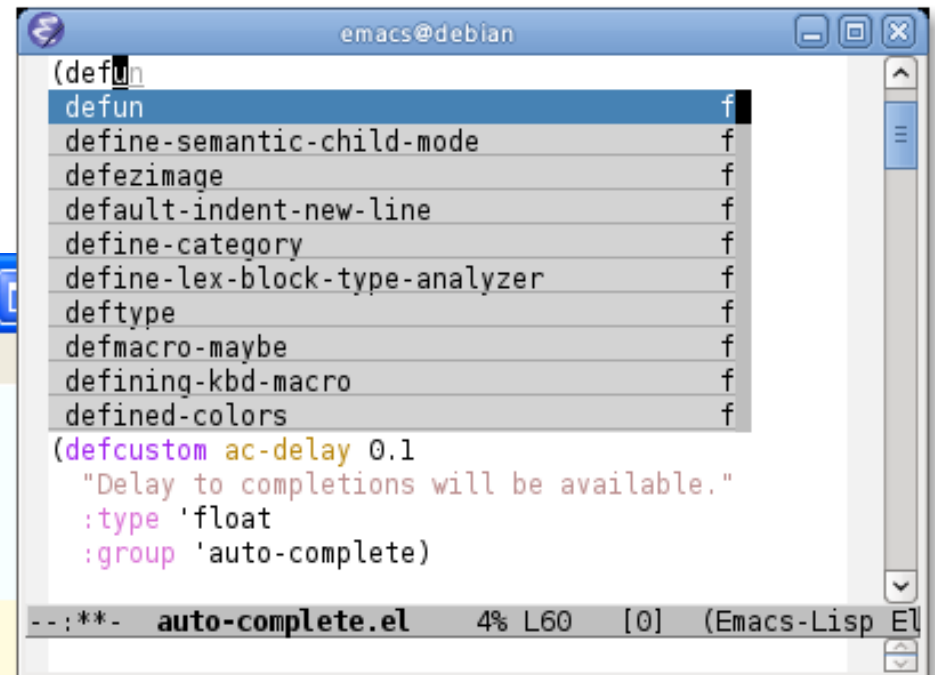
- Mode = режим редактирования текста
- Minor & Major modes
 - php-mode



```
nxhtml-edit.php - Emacs
File Edit Options Buffers Tools PHP C Web Site Help
<?php
require_once('admin.php');
$title = __('Reading Options');
$parent_file = 'options-general.php';
include('admin-header.php');
?>

<div class="wrap">

  <h2><?php _e('Reading Options') ?></h2>
  <form name="form1" method="post" action="options-
ns.php">
    <?php wp_nonce_field('update-options') ?>
--\-- nxhtml-edit.php Top L4 (PHP MuMaMo w32 L A
```



```
emacs@debian
(defun
defun f
define-semantic-child-mode f
defezimage f
default-indent-new-line f
define-category f
define-lex-block-type-analyzer f
deftype f
defmacro-maybe f
defining-kbd-macro f
defined-colors f
(defcustom ac-delay 0.1
  "Delay to completions will be available."
  :type 'float
  :group 'auto-complete)
--:**- auto-complete.el 4% L60 [0] (Emacs-Lisp E
```

* auto-complete-mode



Major mode

- Синтаксическая таблица
(СИМВОЛ : СИНТАКСИЧЕСКИЙ ТИП)
 - *(modify-syntax-entry ?\t " " table)*
 - *(modify-syntax-entry ?\] ")[" table)*
- Сочетания клавиш
(global-set-key "\C-f" 'forward-char)
(global-set-key "\C-M-f" 'my-function)
- Пункты меню

Области видимости

Emacs Lisp

(Dynamic scoping)

```
int x = 1;
```

```
void g () { print (x); x = 2; }
```

```
void f () { int x = 3; g(); }
```

```
f() → 3
```

```
print (x) → 2
```

Common Lisp

(Lexical scoping)

```
int x = 1;
```

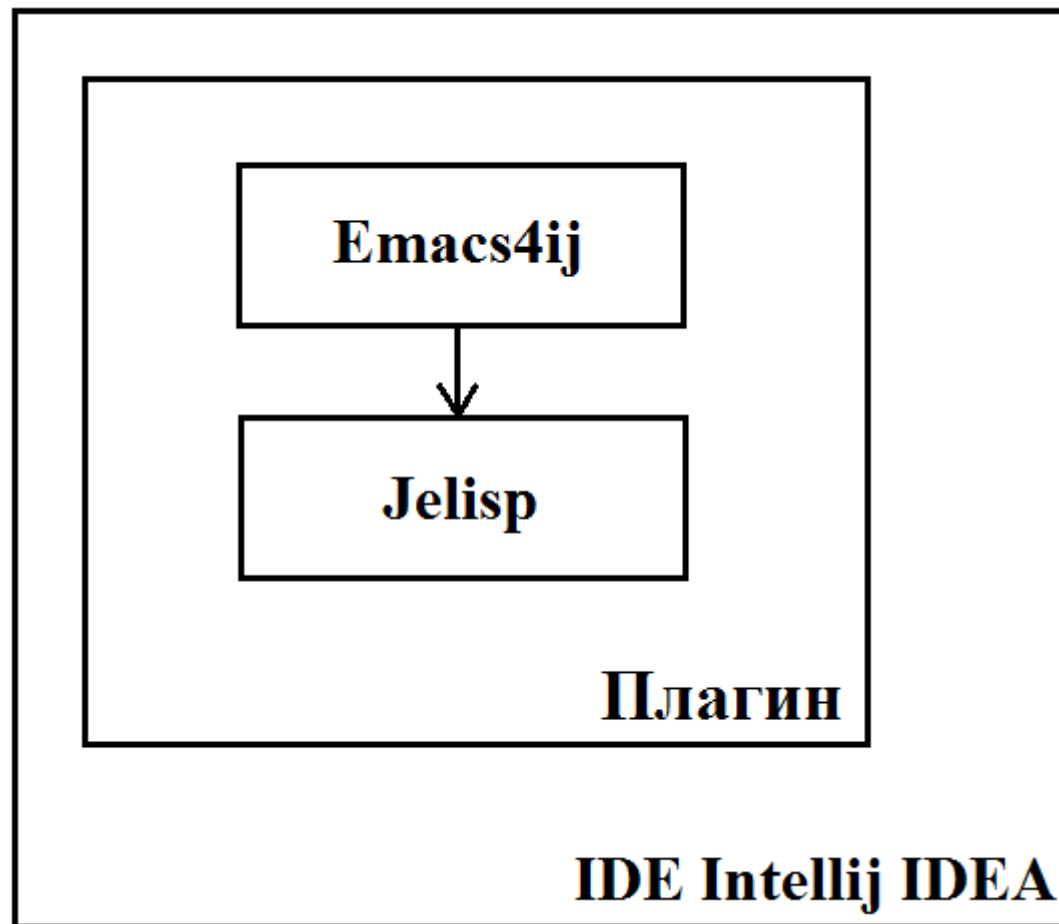
```
void g () { print (x); int x = 2; }
```

```
void f () { int x = 3; g(); }
```

```
f() → 1
```

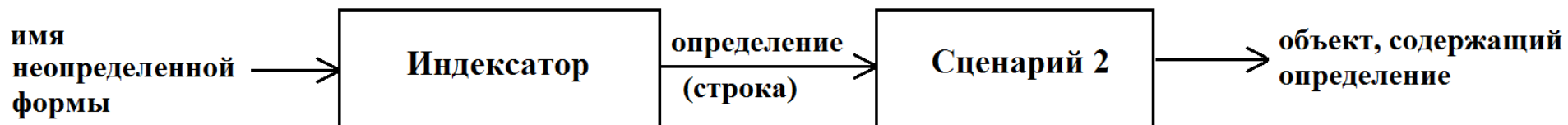
```
print (x) → 1
```

Модульная структура разработанного плагина



Взаимодействие компонентов плагина

(1) Подгрузка определения Emacs Lisp формы



(2) Вычисление значения строки кода на Emacs Lisp

