# Thorn language: a flexible tool for code generation

Yuri Okulovsky
Ural State University

# Domain specific languages

How to create generators for software patterns?

- Write them manually
- Extend the programming language
- Create DSL

# Thorn approach

- New *language* is not required
- New *semantics* is often enough
- Simple syntax
- Easy way to introduce semantics

# Thorn syntax

```
<table border=1 width=100%>
  <tr>
     <td> a11 </td> <td> a12 </td>
  </tr>
  <tr>
     <td> a21 </td> <td> a22 </td>
  </tr>
</table>
```

# Thorn syntax

```
\table[border=1 width=100%] {
  \tr {
     \td {a11} \td{a12}
     }
  \tr {
     \td {a21} \td{a22}
     }
}
```

# Thorn syntax

```
\table[1 100%] {
  \tr {
     \td {a11} \td{a12}
     }
  \tr {
     \td {a21} \td{a22}
     }
}
```

# Thorn syntax

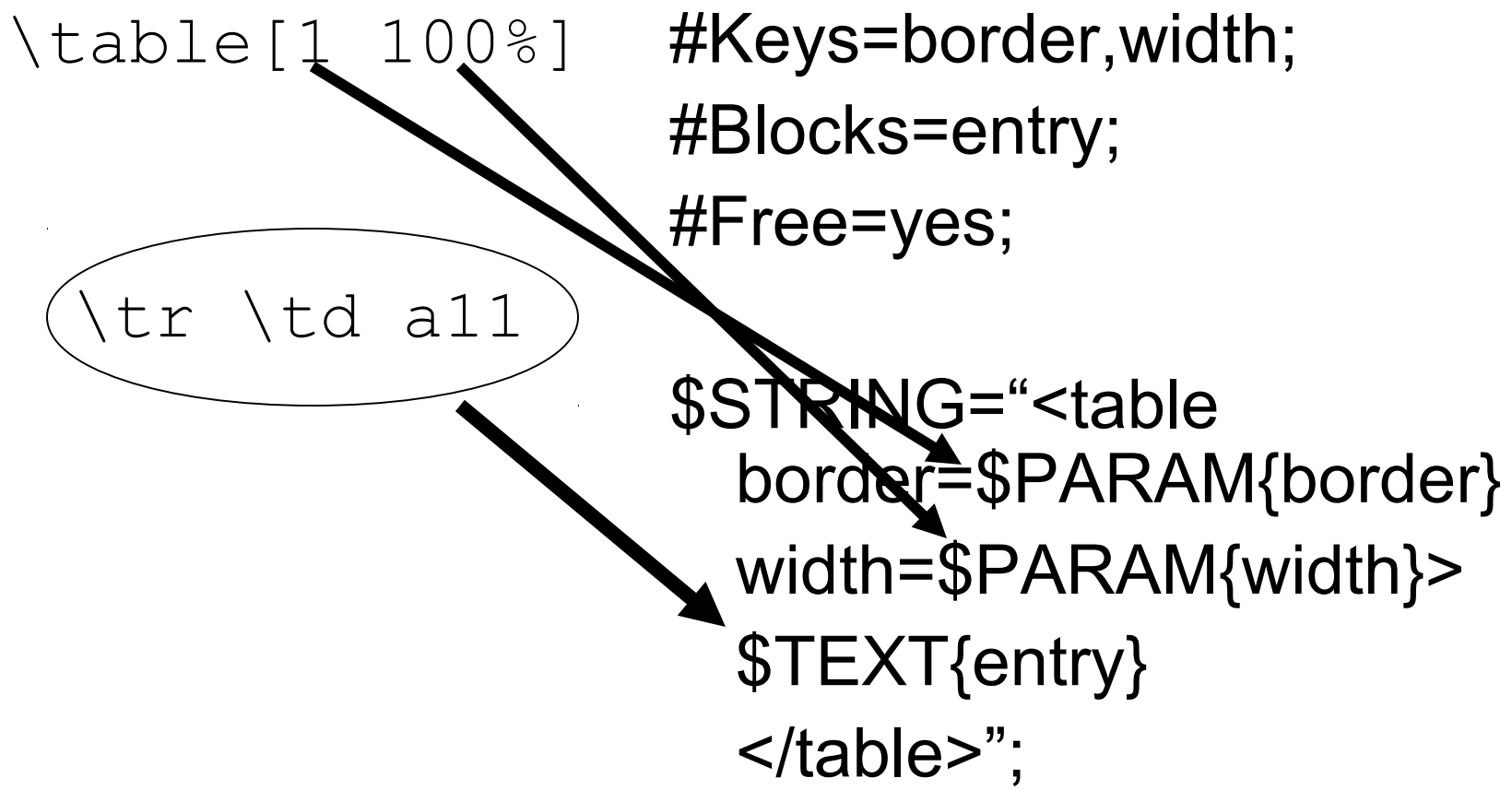
```
\table[1 100%]
  \tr
  \td a11 \td a12

  \tr
  \td a21 \td a22
```

# Thorn semantics

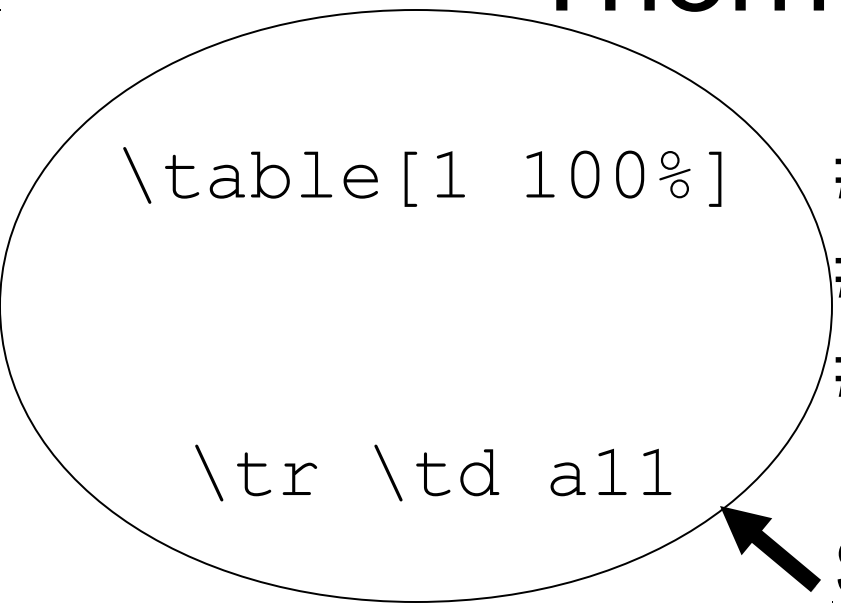
```
\table[1 100%]
  \tr \td a11
```

```
#Keys=border,width;
#Blocks=entry;
#Free=yes;

$STRING="<table
   border=$PARAM{border}
   width=$PARAM{width}>
   $TEXT{entry}
   </table>";
```

# Thorn semantics



# Thorn semantics




#Keys=border,width;
#Blocks=entry;
#Free=yes;

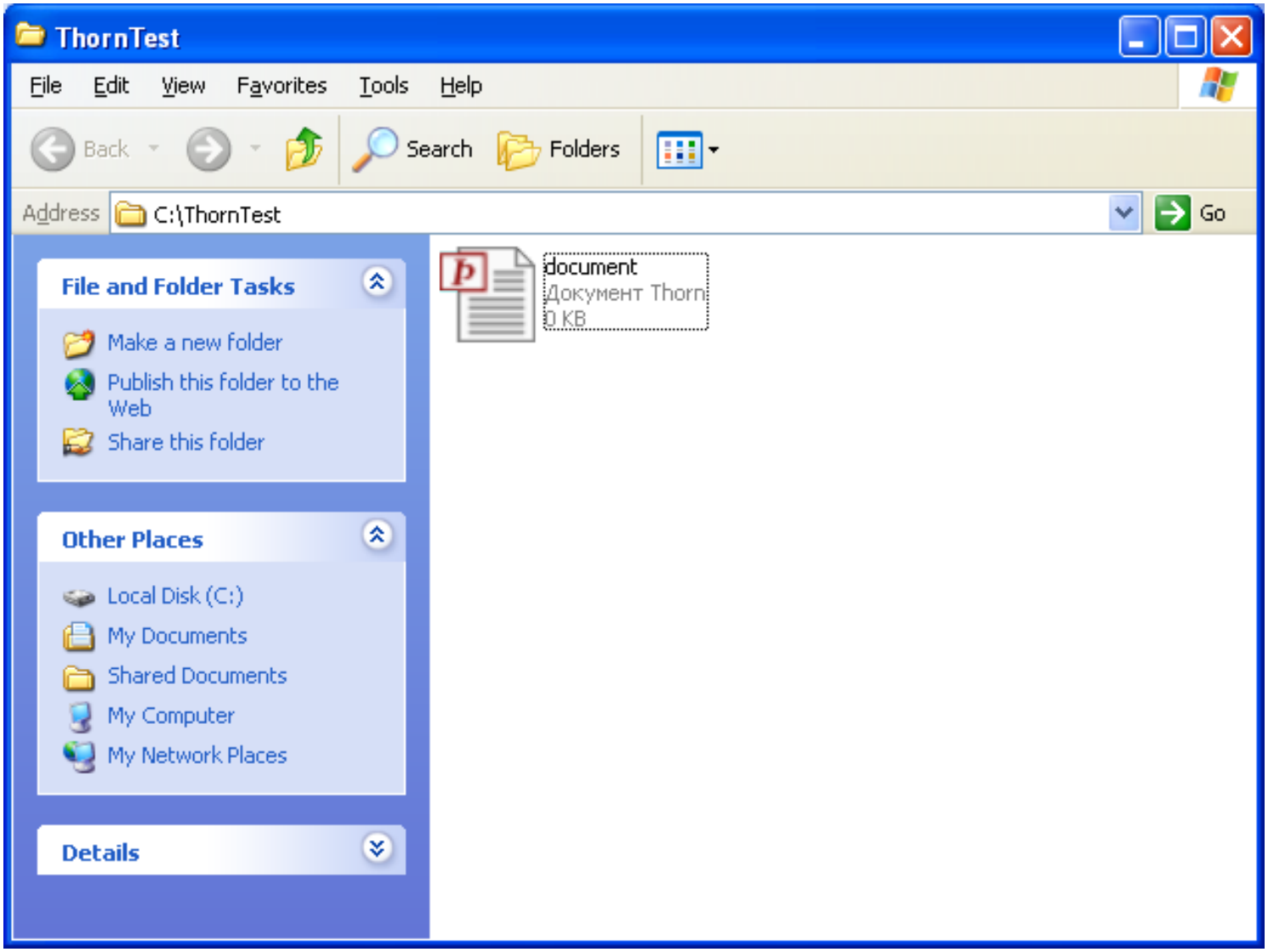
$STRING="<table
border=$PARAM{border}
width=$PARAM{width}>
$TEXT{entry}
</table>";

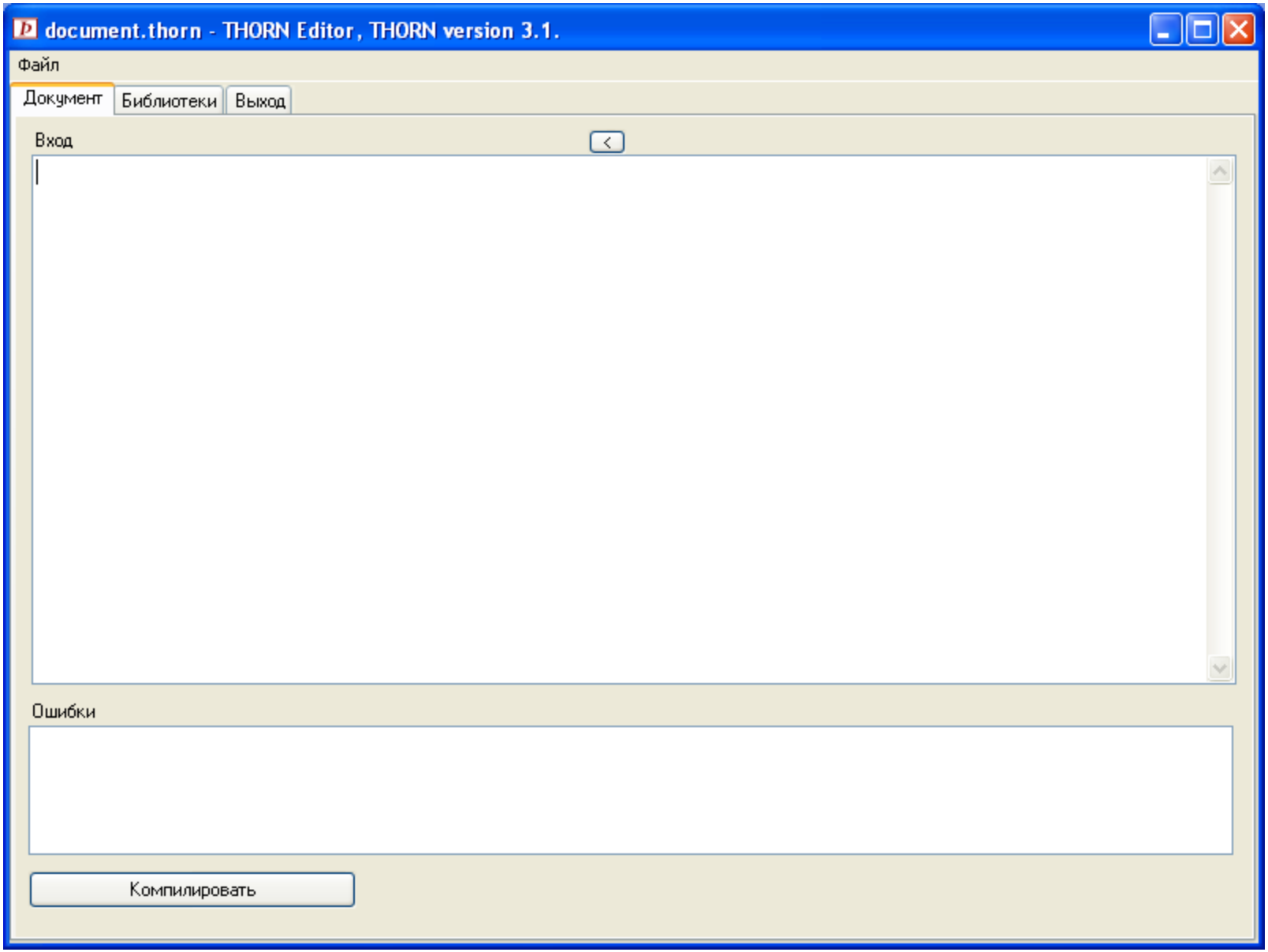
# Thorn semantics

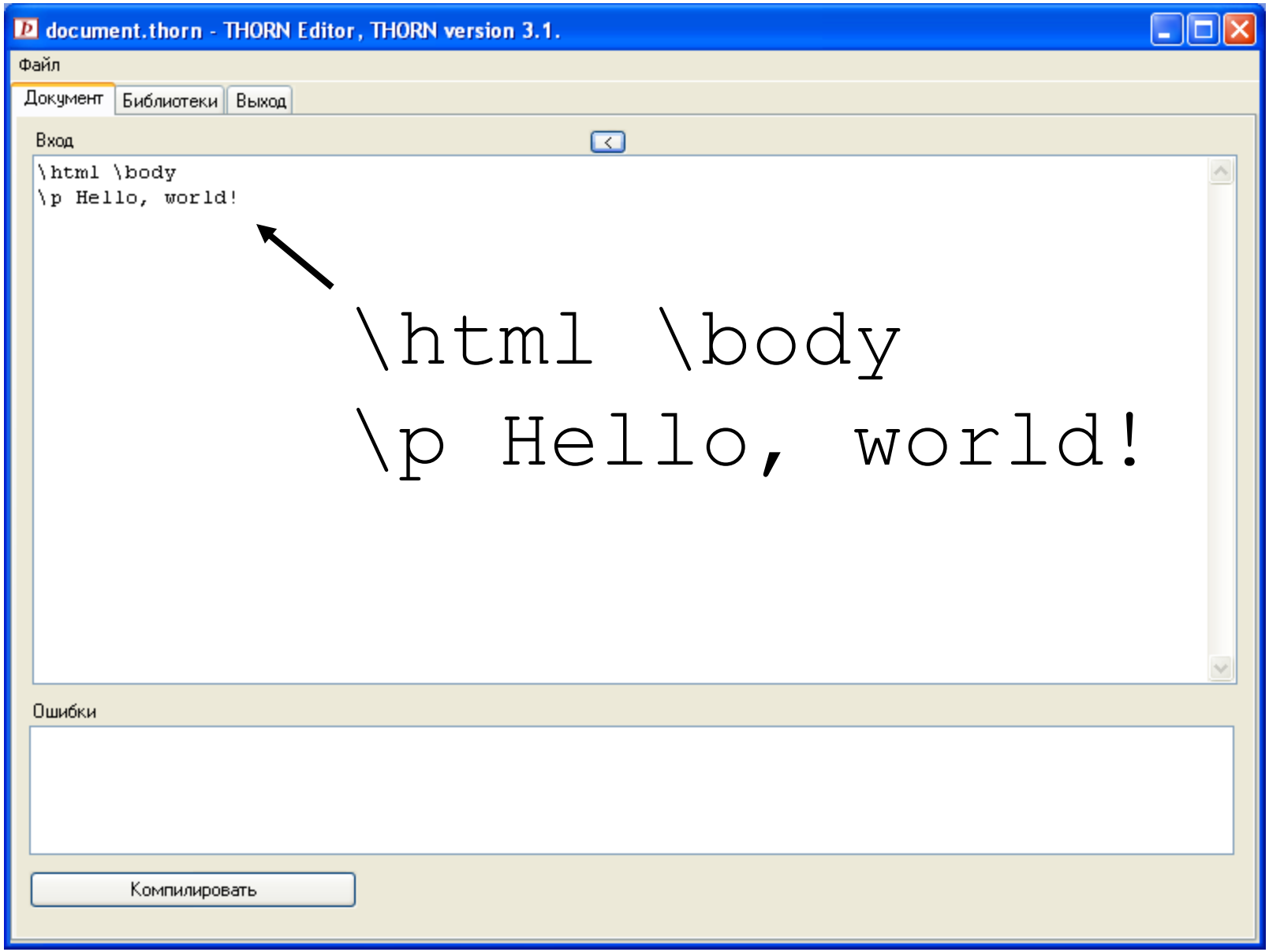
```
#Type=Perl; Keys=Type,Name;


$STRING.="
public event
    $PARAM{Type}EventHandler
    $PARAM{Name};
```

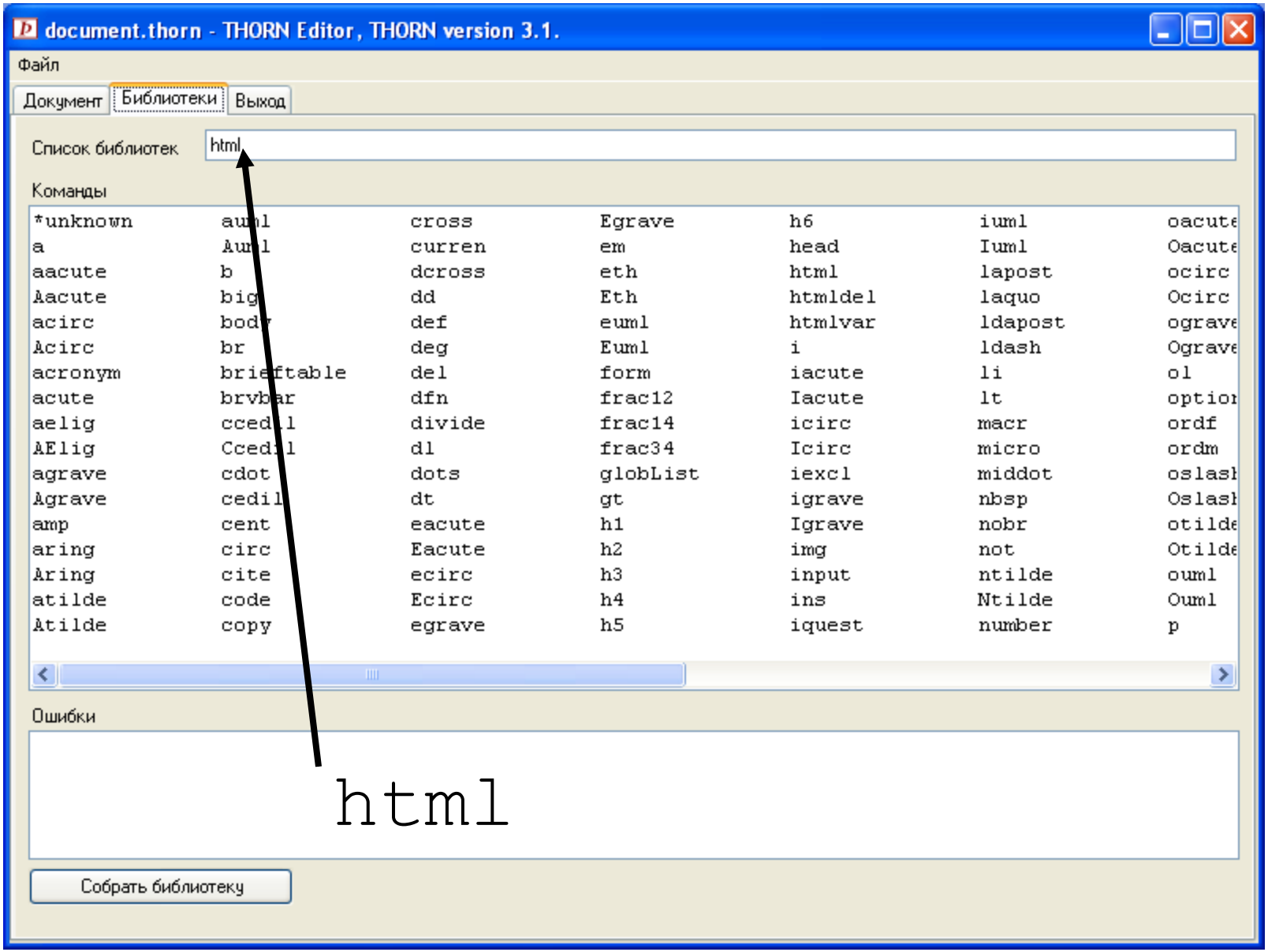
# Thorn semantics

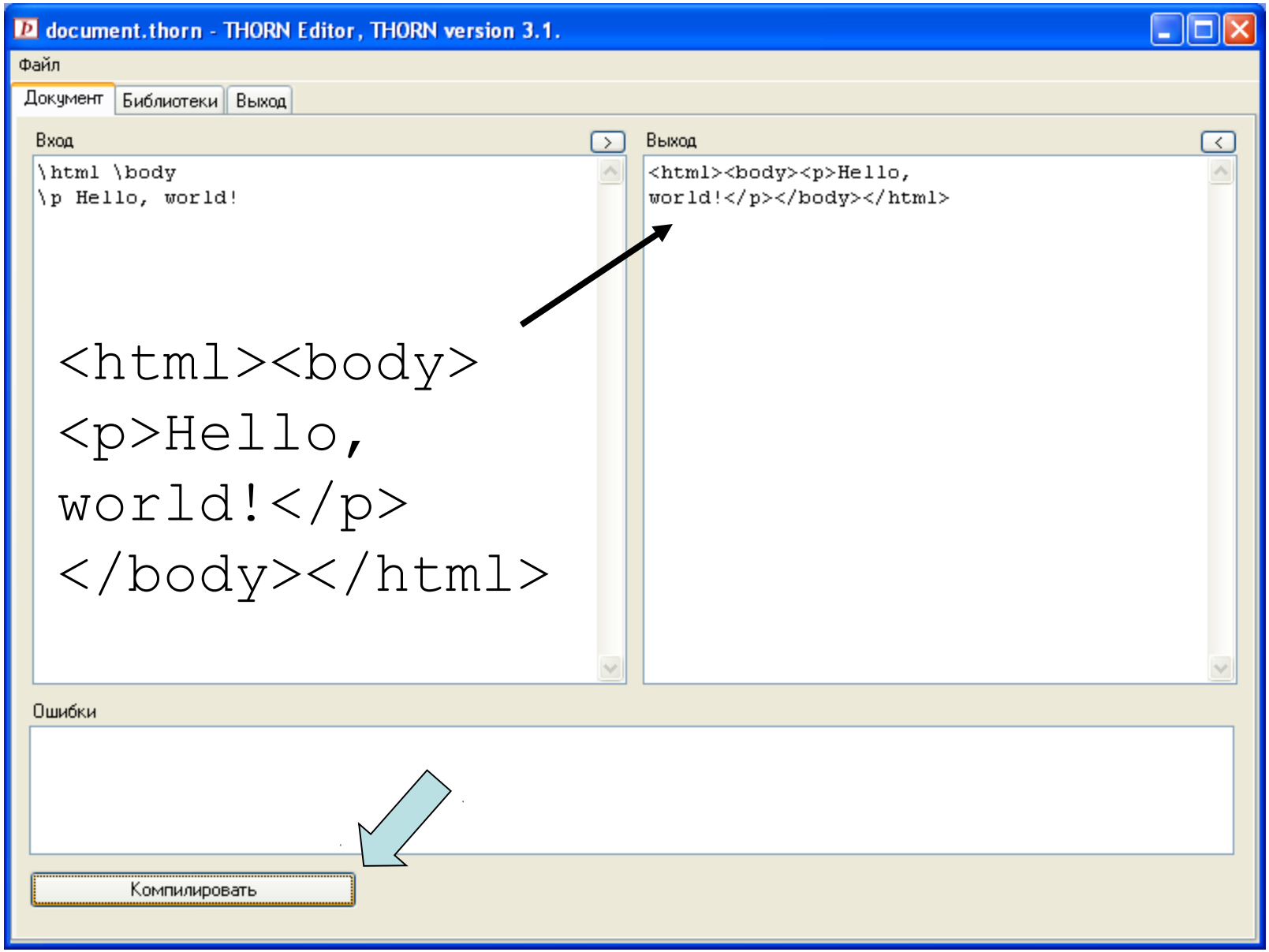
```
Protected void On$PARAM{Name}(
  $PARAM{Type}EventArgs args)
  {
  if ($PARAM{Name}!=null)
     $PARAM{Name}(this,args);
  }
```

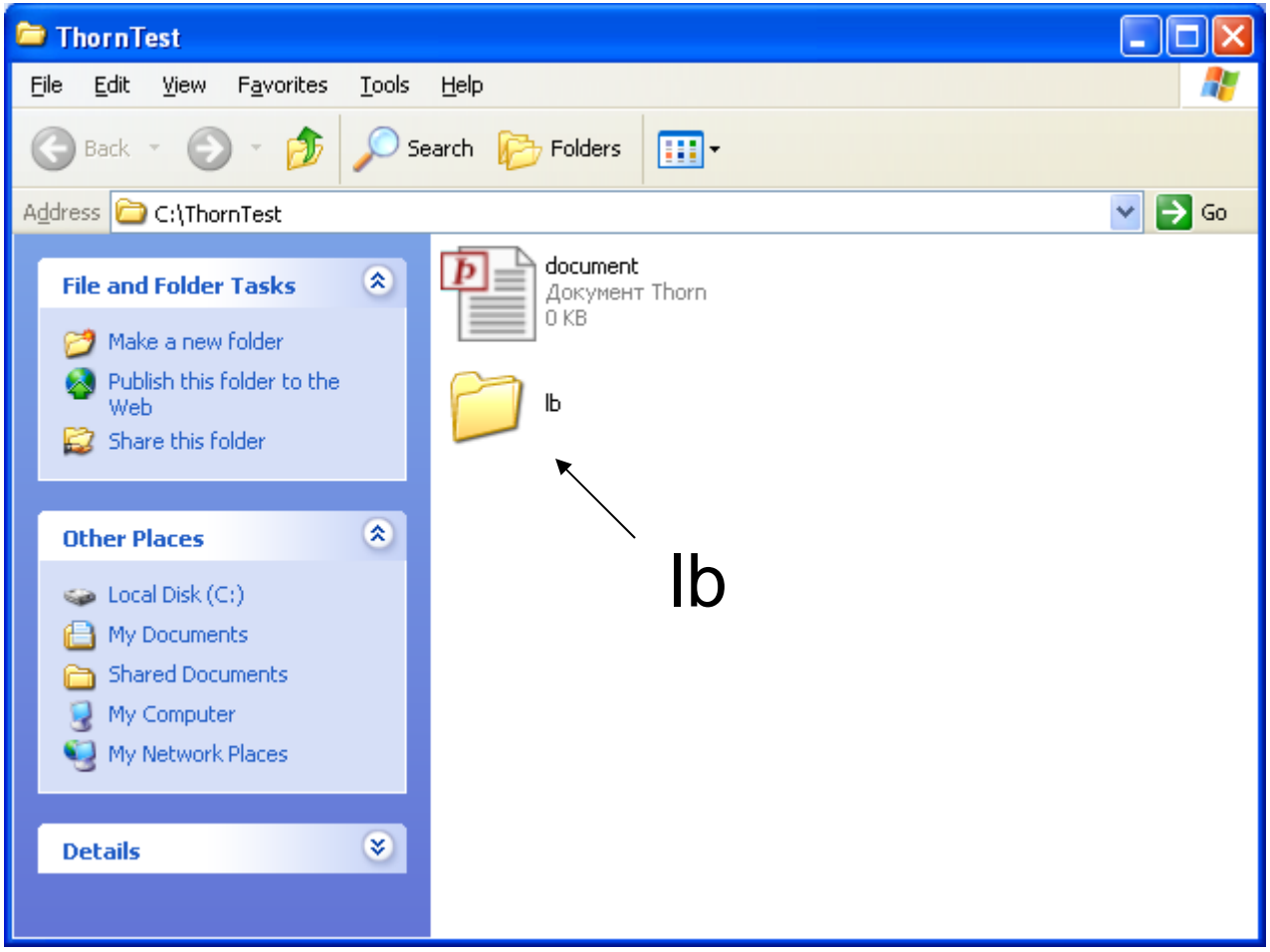

ThornTest
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\ThornTest
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
Other Places
Local Disk (C:)
My Documents
Shared Documents
My Computer
My Network Places
Details
document
Документ Thorn
0 KB

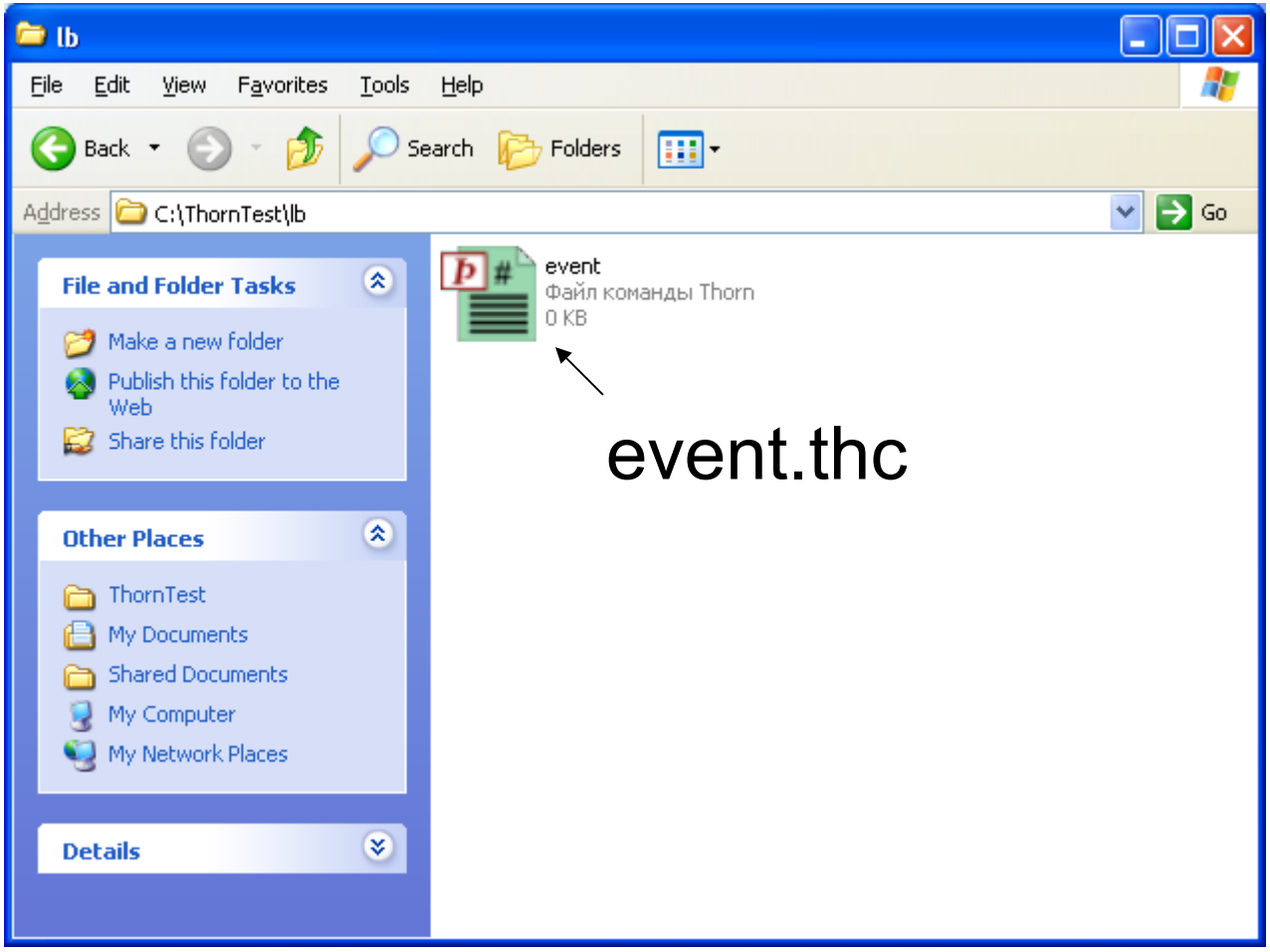

document.thorn - THORN Editor, THORN version 3.1.
Файл
Документ
Библиотеки
Выход
Вход
Ошибки
Компилировать

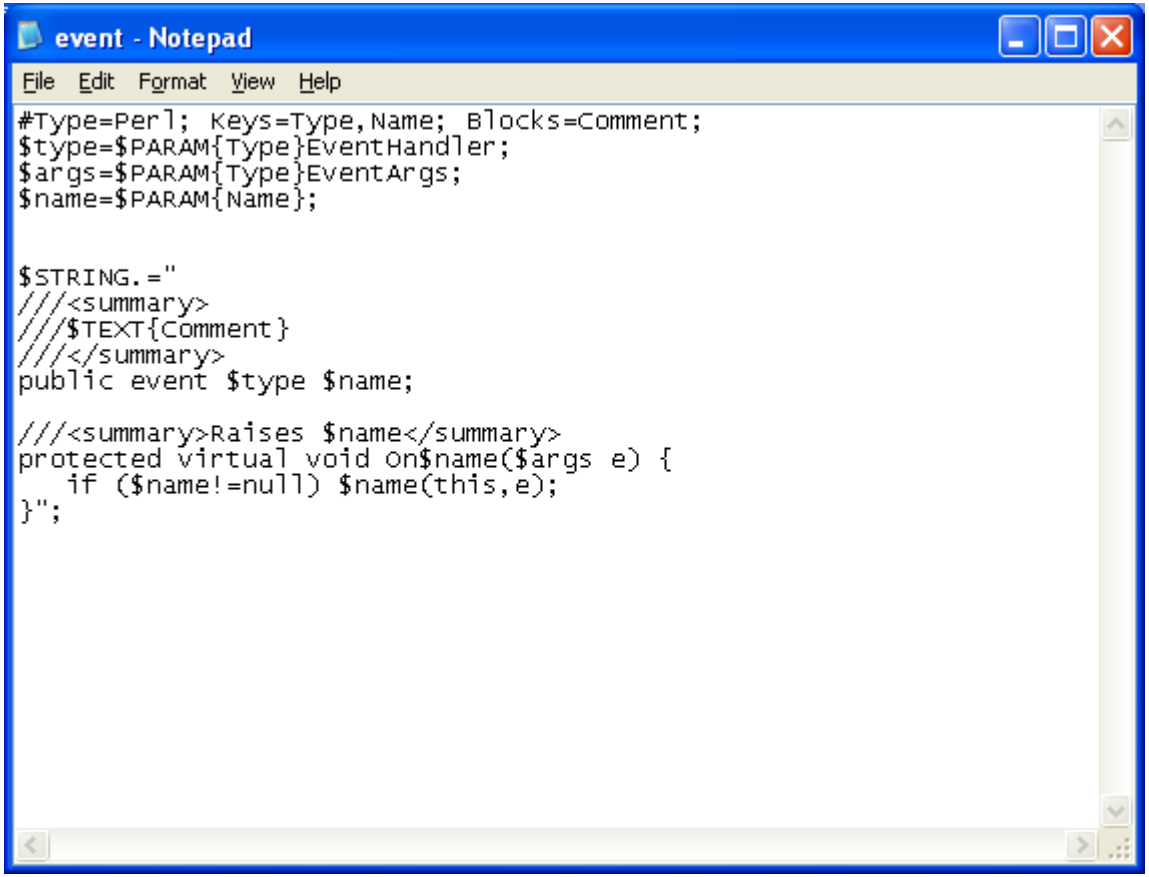

document.thorn - THORN Editor, THORN version 3.1.
Файл
Документ
Библиотеки
Выход
Вход
\html \body
\p Hello, world!
\html \body
\p Hello, world!
Ошибки
Компилировать

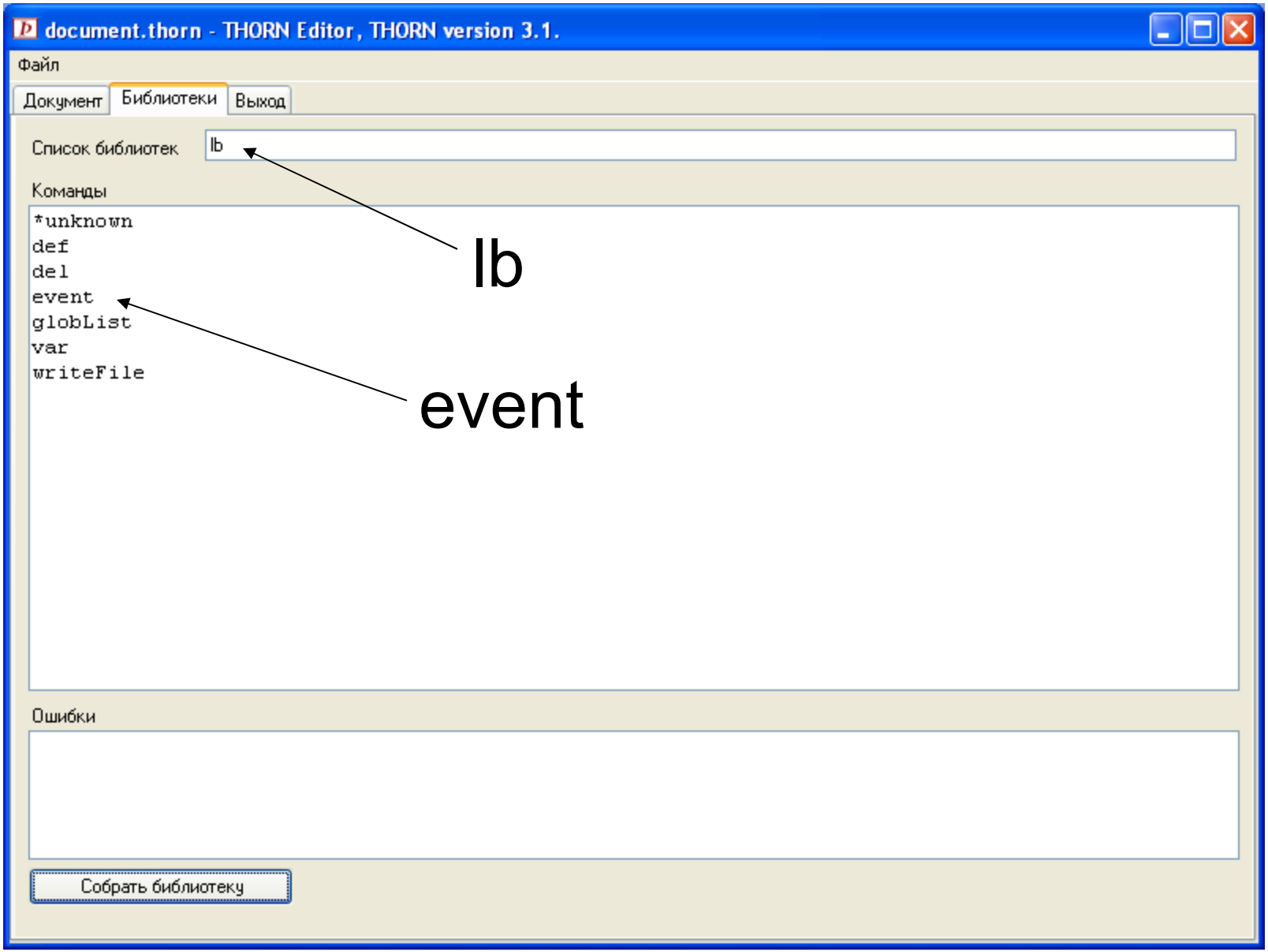

document.thorn - THORN Editor, THORN version 3.1.
Файл
Документ
Библиотеки
Выход
Список библиотек
html
Команды
*unknown auml cross Egrave h6 iuml oacute
a Auml curren em head Iuml Oacute
aacute b dcross eth html lapost ocirc
Aacute big dd Eth htmldel laquo Ocirc
acirc bodv def euml htmlvar ldapost ograve
Acirc br deg Euml i ldash Ograve
acronym brieftable del form iacute li ol
acute brvbar dfn frac12 Iacute lt optio
aelig ccedil divide frac14 icirc macr ordf
AElig Ccedil dl frac34 Icirc micro ordm
agrave cdot dots globList iexcl middot oslash
Agrave cedil dt gt igrave nbsp Oslash
amp cent eacute h1 Igrave nobr otilde
aring circ Eacute h2 img not Otilde
Aring cite ecirc h3 input ntilde ouml
atilde code Ecirc h4 ins Ntilde Ouml
Atilde copy egrave h5 iquest number p
Ошибки
Собрать библиотеку
html

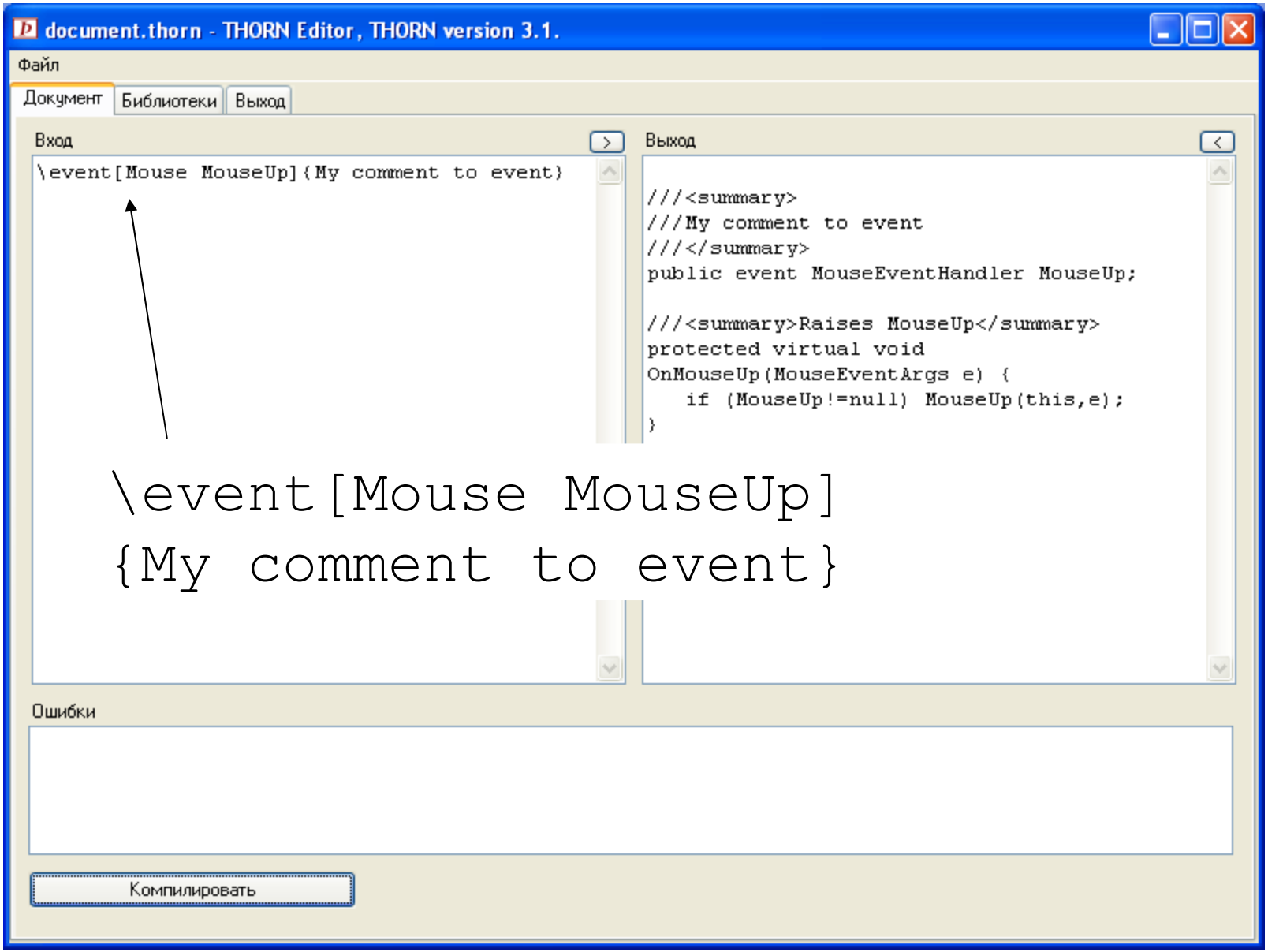

document.thorn - THORN Editor, THORN version 3.1.
Файл
Документ
Библиотеки
Выход
Вход
\html \body
\p Hello, world!
Выход
<html><body> Hello,
world! </body></html>
<html><body>
Hello,
world!
</body></html>
Ошибки
Компилировать

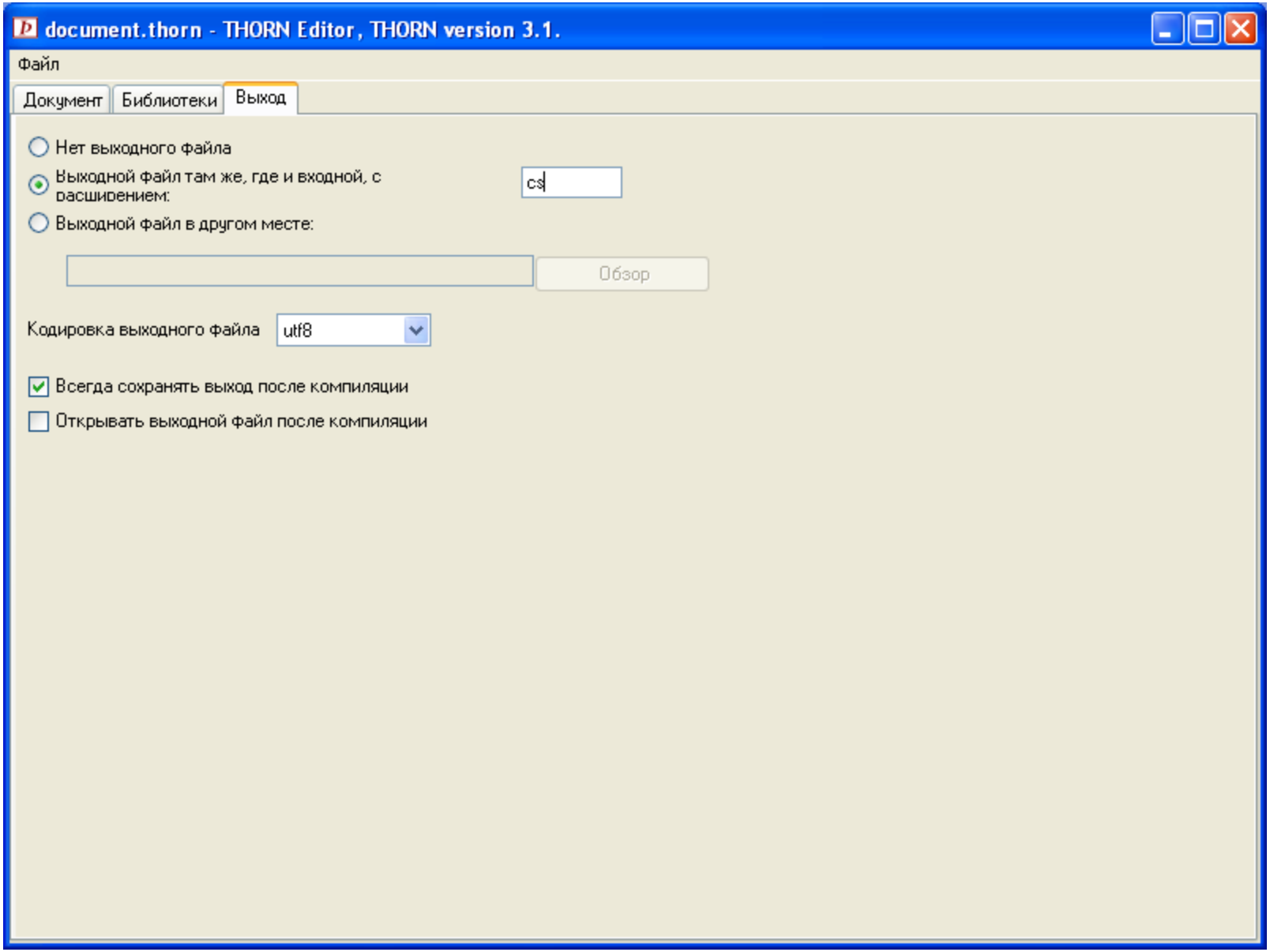

ThornTest
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\ThornTest
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
Other Places
Local Disk (C:)
My Documents
Shared Documents
My Computer
My Network Places
Details
document
Документ Thorn
0 KB
lb
lb


lb
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\ThornTest\lb
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
Other Places
ThornTest
My Documents
Shared Documents
My Computer
My Network Places
Details
event
Файл команды Thorn
0 KB
event.thc


event - Notepad
File Edit Format View Help
#Type=Perl; Keys=Type,Name; Blocks=Comment;
$type=$PARAM{Type}EventHandler;
$args=$PARAM{Type}EventArgs;
$name=$PARAM{Name};
$STRING.="
///<summary>
///$TEXT{Comment}
///</summary>
public event $type $name;
///<summary>Raises $name</summary>
protected virtual void On$name($args e) {
if ($name!=null) $name(this,e);
}";


document.thorn - THORN Editor, THORN version 3.1.
Файл
Документ
Библиотеки
Выход
Список библиотек
lb
Команды
*unknown
def
del
event
globList
var
writeFile
Ошибки
Собрать библиотеку
lb
event


document.thorn - THORN Editor, THORN version 3.1.
Файл
Документ
Библиотеки
Выход
Вход
\event[Mouse MouseUp]{My comment to event}
Выход
///<summary>
///My comment to event
///</summary>
public event MouseEventHandler MouseUp;
///<summary>Raises MouseUp</summary>
protected virtual void
OnMouseUp(MouseEventArgs e) {
if (MouseUp!=null) MouseUp(this,e);
}
\event[Mouse MouseUp]
{My comment to event}
Ошибки
Компилировать


document.thorn - THORN Editor, THORN version 3.1.
Файл
Документ
Библиотеки
Выход
Нет выходного файла
Выходной файл там же, где и входной, с расширением:
cs
Выходной файл в другом месте:
Обзор
Кодировка выходного файла
utf8
Всегда сохранять выход после компиляции
Открывать выходной файл после компиляции

# Fungi

```
\bibliography
    \item[book]
        \author John Smith
        \title My book
    ...
\print
```

# Fungi

\item
    \title On animals
    \author John Smith
\item
    \title On flowers
    \author Jane Smith

| Title | Authors |
| --- | --- |
| On animals | John Smith |
| On flowers | Jane Smith |
| | |

# Fungi

```
#Blocks=Author;Free=yes;
#Parents=item;Type=Perl;

require 'db.pm';
$db=db->new(\%GLOBAL);
$db->SetFieldInCurrentRow
     ("Authors",$TEXT{Author});
```

# Fungi

#Blocks=Author;Free=yes;

#Parents=item;**Type=FungiSetter;**

# Fungi

```
#Type=Perl;
require 'db.pm';
sub MakeItem {
    %h=@_;
    $STRING.="$h{Author}. $h{Name}\n";
    }
$db=db->new(\%GLOBAL);
$db->RunOver(\&MakeItem);
```

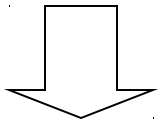
# Thornado

- An environment to build data-driven business application
- Consists of Thorn library and .NET auxiliary assembly
- Allows describing data on Thorn and then generate SQL queries, GUI, etc.

# Thornado

- Simplifies reflection structure:

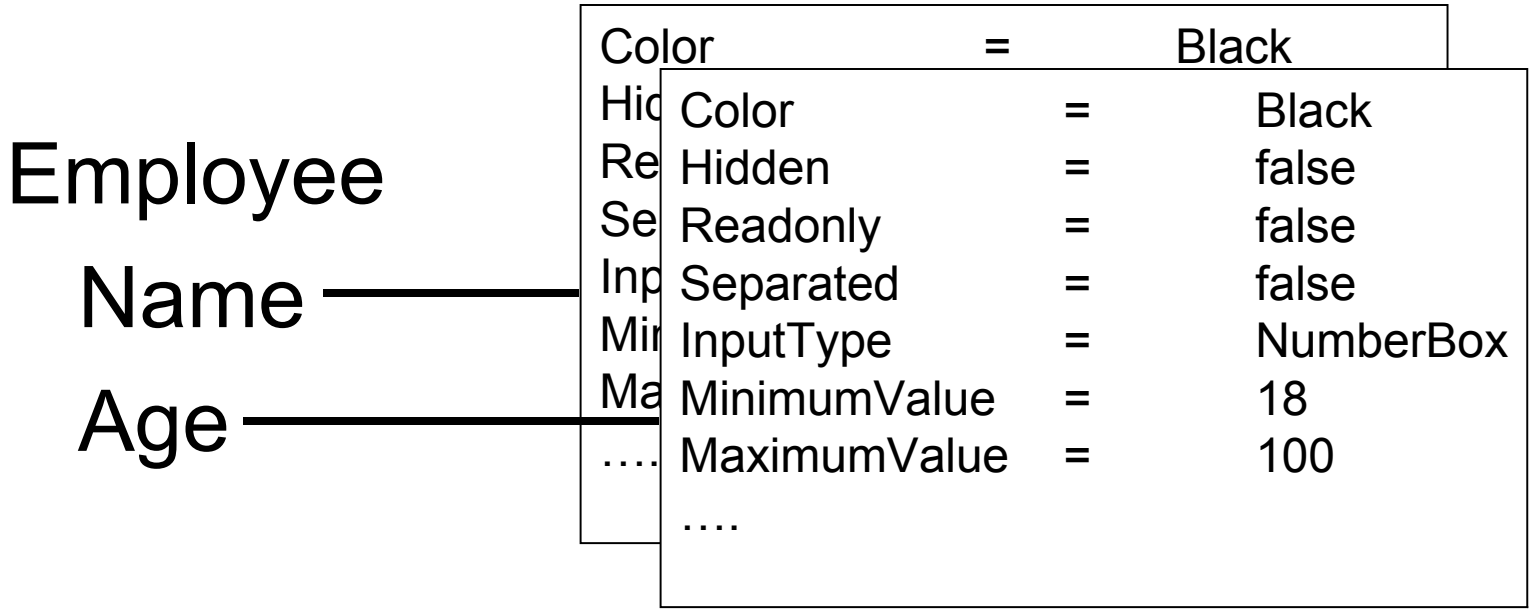
Department[“Math”].Employees[23].Name



Department.Math.Employees.23.Name

# Thornado

- Binds information to data



# Thornado

- Advanced business-logic check
  - Checking the data correctness
  - Storing the list of errors
  - Data changes the bound data

# Thornado

- WinForms graphic user interface
- Web-based graphic user interface **[underconstruction]**

# Thornado

```
\field[string Email]
   \desc E-mail address
      \io TypeIO.String
   \gui Text
   \check
      \err.W v==""
      -- Address must be entered
      \err.W v.Length<5
      -- Address is too short
      if (!v.Contains('@'))
          list.Add("Address is not valid");
```

Thank you. Your questions?