

НТУ «ХПИ»  
Кафедра АСУ

**Шаблон**  
Для оформления учебной документации  
с использованием издательской системы  $\text{\LaTeX}$

Автор  
Livich: <https://github.com/Livich/>  
На основе разработок Vbolshutkin:  
<https://github.com/vbolshutkin>

Харьков 2013

# ЗМІСТ

1	Введение.....	3
1.1	Лицензия.....	3
1.2	Принципы использования .....	3
1.3	Литература.....	4
2	Документация опциональных пакетов.....	5
2.1	Опциональный пакет «kpi.util.image» .....	5
2.1.1	Макрос «putImage» .....	5
2.1.2	Известные проблемы .....	5
2.2	Опциональный пакет «kpi.util.listing» .....	5
2.2.1	Макрос «putListing» .....	5
2.2.2	Макрос «listingLink» .....	6
2.2.3	Известные проблемы .....	6
2.3	Опциональный пакет «kpi.util.draft».....	6
2.3.1	Макрос «ifxdraft» .....	7
2.3.2	Макрос «dPutImage» .....	7
2.3.3	Макрос «dPutListing» .....	7
2.3.4	Макрос «draftPage» .....	7
2.4	Опциональный пакет «kpi.title.coursework» .....	7
2.4.1	Макрос «maketitle» .....	8
2.5	Опциональный пакет «kpi.title.report» .....	8
2.5.1	Макрос «maketitle» .....	8
3	Примеры использования .....	9
3.1	Опциональный пакет «kpi.util.image» .....	9
3.2	Опциональный пакет «kpi.util.listing» .....	9
3.3	Опциональный пакет «kpi.title.coursework» .....	12

## 1 ВВЕДЕНИЕ

### 1.1 Лицензия

Данный набор шаблонов выпускается под лицензией CC-BY-SA (<http://creativecommons.org/licenses/by-sa/3.0/>), то есть никаких гарантий на функционал или просто работоспособность не предоставляется. Вы можете как угодно изменять этот пакет, с сохранением первоначального авторства за его оригинальными авторами (Livich и Vbolshutkin).

Этот набор не является официальным и не поддерживается никаким подразделением НТУ «ХПИ». Любые справки могут быть получены только непосредственно в репозитории проекта на GitHub: [https://github.com/Livich/stvuz\\_KhPI\\_XeLaTeX](https://github.com/Livich/stvuz_KhPI_XeLaTeX).

### 1.2 Принципы использования

Этот пакет шаблонов используется так же, как и любой другой пакет для  $\text{\LaTeX}$ , однако сам набор поделён на несколько частей: одну обязательную (базовый пакет) и множество необязательных, обеспечивающих дополнительный функционал.

В пакете принято соглашение об именовании файлов так, чтобы можно было получить доступ к части пакета интуитивно. Базовый пакет пока не подчинён этому соглашению.

Таким образом, используя данный набор, необходимо придерживаться следующей схемы:

- 1) Установить и настроить компилятор  $\text{\LaTeX}$  на вашем компьютере, если это не сделано ранее;
- 2) Установить и настроить  $\text{BibTeX}$  на вашем компьютере, если это не сделано ранее;
- 3) Установить все зависимые пакеты, если это не сделано ранее;
- 4) Распаковать данный набор в доступную для  $\text{\LaTeX}$ -компилятора директорию, если это не сделано ранее;
- 5) Подключить базовый пакет к вашему документу или взять за основу готовый шаблон документа из этого пакета (`document.tex`);
- 6) Опционально подключить дополнительные пакеты из набора.

Именованье опциональных пакетов производится следующим образом:

- Если это пакет-утилита (произвольный дополнительный функционал), то он именуется как **kpi.util.<идентификатор>.sty**;
- Если это пакет-титульная страница, то он именуется как **kpi.title.<идентификатор>.sty**.

Других правил именования пока не предусмотрено.

### 1.3 Литература

Об установке, настройке и использовании  $\text{\LaTeX}$ ,  $\text{\XeTeX}$ ,  $\text{\BibTeX}$  написано достаточно большое количество материалов, которые могут пригодиться, если вы впервые работаете с этой издательской системой.

Полезными окажутся:

- Книга, С.М. Львовский: «Набор и вёрстка в системе  $\text{\TeX}$ »;
- Веб-сайт, <http://mydebianblog.blogspot.com/2009/04/miktex-windows.html>;
- Веб-сайт, <http://www.tug.org/texlive/>.

## 2 ДОКУМЕНТАЦИЯ ОПЦИОНАЛЬНЫХ ПАКЕТОВ

### 2.1 Опциональный пакет «kpi.util.image»

Обеспечивает поддержку изображений. Макросы:

1) putImage.

#### 2.1.1 Макрос «putImage»

putImage позволяет отобразить изображение с подписью. Использование:

```
\putImage{[имя файла изображения]}\{[подпись]}\{[масштаб]}\{[метка]}
```

При этом будет создана метка «[метка]», указывающая на изображение. Изображение будет подписано и пронумеровано.

#### 2.1.2 Известные проблемы

Согласно стандарту, перед и после объекта изображения должны располагаться отступы, однако, в случае последовательного расположения двух и более изображений, пропуски удваиваться не должны.

Пока нормальным поведением для пакета является удваивание пропусков в таких случаях. Для того чтобы не нарушать стандарт, следует избегать последовательной вставки рисунков.

### 2.2 Опциональный пакет «kpi.util.listing»

Обеспечивает поддержку листингов.

Макросы:

1) putListing;

2) listingLink.

Внимание: пакет переопределяет настройки пакета listings и переопределяет команды пакета framed.

Зависит от пакетов: listings, framed (используется модифицированный пакет из комплекта поставки)

#### 2.2.1 Макрос «putListing»

putListing вставляет листинг с разбивкой на страницы, как будто это изображения. Использование:

```
\putListing{[исходный файл]}\{[подпись]}\{[метка]}\{[язык листинга]}
```

При этом будут созданы label «[метка]-0»,..., «[метка]-N», «[метка]-last», которые последовательно указывают на первую и последующие страницы листинга. Label «-0» не будет определена, если листинг не разбивался, вместо неё определяется сразу «-last» и это будет единственная метка, указывающая на этот листинг.

Также будет создан счётчик «[метка]:counter», содержащий количество страниц в листинге, однако соответствующие действительному числу страниц листинга значения он принимает только после того, как все части листинга были вставлены в документ. Счётчик принимает значения, равные общему числу страниц листинга, включая страницы, не полностью заполненные листингом.

### 2.2.2 Макрос «listingLink»

listingLink вставляет ссылку на листинг, учитывая разбивку. Так, если листинг занимает только один рисунок, то будет возвращена ссылка на только один этот рисунок. Если рисунков больше - это тоже учитывается и будут возвращены ссылки на первый и последний рисунки через дефис. Использование:

```
\listingLink{[метка листинга]}
```

listingLink возвращает значения без какого-либо предваряющего или завершающего пробела.

### 2.2.3 Известные проблемы

Пакет framed, использующийся для обрамления листингов в рамки используется в модифицированном виде. Это связано с тем, что в стандартной имплементации framed не позволяет добавить подпись к объекту в рамке так, чтобы он умещался на той же странице, что и объект в рамке.

Пакет framed модифицирован так, что обрамлённый в рамку объект дополняется значением команды theFramedSubCaption, которая определяется в «kpi.util.listing».

В настоящий момент проблема с использованием модифицированного пакета не решена и последняя просто включена в комплект поставки набора.

## 2.3 Опциональный пакет «kpi.util.draft»

Обеспечивает поддержку режима черновика. Макросы:

- 1) ifxdraft;
- 2) dPutImage;

3) dPutListing;

4) draftPage.

Зависит от пакетов: datetime, ifdraft

### 2.3.1 Макрос «ifxdraft»

Позволяет выполнять проежку режима черновика. Использование:

```
\ifxdraft [если черновик] \else [если не черновик] \fi
```

### 2.3.2 Макрос «dPutImage»

Позволяет отобразить вместо изображения его заглушку в режиме черновика, и само изображение в чистовом режиме. Использование:

```
\dPutImage{[имя файла изображения]}\{[подпись]}\{[масштаб]}\{[метка]}
```

Использование макроса аналогично putImage.

### 2.3.3 Макрос «dPutListing»

Позволяет отобразить вместо листинга его заглушку в режиме черновика, и сам листинг в чистовом режиме. Использование:

```
\dPutListing{[исходный файл]}\{[подпись]}\{[метка]}\{[язык листинга]}
```

Использование макроса аналогично putListing.

### 2.3.4 Макрос «draftPage»

Позволяет подключать к документу-черновику страницу с информацией о версии документа. Использование:

```
\draftPage
```

В результате использования будет отображена страница, извещающая о том, что этот документ является черновиком. Также будет показана таблица со сводной информацией о версии документа.

В случае, если документ компилируется в чистовом режиме, команда не делает ничего.

## 2.4 Опциональный пакет «kpi.title.coursework»

Обеспечивает генерацию титульной страницы для курсовой работы.

Зависимости: hyperref.

Внимание: подключение пакета изменяет некоторые настройки hyperref.

Макросы:

1) maketitle.

### 2.4.1 Макрос «maketitle»

makeTitle генерирует титульную страницу для курсовой работы. Использование:

`\maketitle`

maketitle требует чтобы были определены следующие макросы:

- 1) docDepartment - название кафедры;
- 2) workSupervisor - руководитель работы, напр. "Сокол В.Е.";
- 3) workSupervisorPosition - статус руководителя работы, например "ассистент каф. АСУ";
- 4) docSubject - название курсовой работы;
- 5) studentGroup - группа автора;
- 6) studentName - ФИО автора;
- 7) docChecker - ФИО проверяющего;
- 8) docCheckerPosition - статус проверяющего, например "голова комісії професор".

## 2.5 Опциональный пакет «kpi.title.report»

Обеспечивает генерацию титульной страницы «Звіт».

Зависимости: `hyperref`.

Внимание: подключение пакета изменяет некоторые настройки `hyperref`.

Макросы:

- 1) `maketitle`.

### 2.5.1 Макрос «maketitle»

makeTitle генерирует титульную страницу «Звіт». Использование:

`\maketitle`

maketitle требует чтобы были определены следующие макросы:

- 1) docNumber - номер (лабораторной) работы;
- 2) docDepartment - название кафедры;
- 3) docSubject - название учебной дисциплины;
- 4) studentGroup - группа автора;
- 5) studentName - ФИО автора;
- 6) docChecker - должность и ФИО проверяющего.



### 3 ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ

#### 3.1 Опциональный пакет «kpi.util.image»

Пакет позволяет вставлять изображения.

Пример приведён на рисунке 3.1.



Рисунок 3.1 – Пример вставки изображения

#### 3.2 Опциональный пакет «kpi.util.listing»

Пакет позволяет вставлять листинги в соответствии со стандартом СТВУЗ, то есть оформлять их как рисунки. Листинг разбивается на множество изображений, каждое изображение нумеруется и подписывается.

Пример вставки листинга, автоматических ссылок и автоматической разбивки его на отдельные рисунки приведён на рисунках 3.2 – 3.5.

```
1 #include "stdafx.h"
2 #include <winsock2.h>
3 #include <stdio.h>
4 #include <windows.h>
5 #include <iostream>
6 #include "..\..\..\lab1\sp_lab1\sp_lab1\matrix.cpp";
7 #pragma comment (lib, "ws2_32.lib")
8
9 #ЛР, системное программирование
10 #Выполнил ст. гр. ИФГ31–
```

Рисунок 3.2 – Исходный код файла «./etc/listing.cpp»

```

11 #Лиликович СА..
12 #2013 год
13
14
15 void TheError(char* message){
16     std::cout<<"ERROR:_"<<message<<"_,_GetLastError="<<GetLastError()<<
17     ",_WSAGetLastError="<<WSAGetLastError()<<"._\n";
18 }
19
20 const int COUNT_OF_MATRICES=2;
21 int _tmain(int argc, _TCHAR* argv[])
22 {
23     matrix*matrices = new matrix[COUNT_OF_MATRICES];
24
25     for(int i=0;i<COUNT_OF_MATRICES;i++){
26         int W, H;
27         std::cout<<"Enter_matrix_"<<i+1<<"_width_and_height:_\n";
28         std::cin>>W>>H;
29         matrices[i] = *(new matrix(W,H));
30         for(int y=1;y<H+1;y++){
31             std::cout<<"Enter_row_"<<y<<"_of_matrix_"<<i+1<<":\n";
32             for(int x=1;x<W+1;x++){
33                 float I = 0;
34                 std::cin>>I;
35                 matrices[i].setval(x,y,I);
36             }
37         }
38         std::cout<<"Got_matrix_"<<i+1<<"_of_"<<COUNT_OF_MATRICES<<":\n";
39         matrices[i].render();
40         std::cout<<"\n";
41     }
42
43     std::cout<<"Got_all_matrices._Start_client_now.\n";
44     system("pause");
45

```

Рисунок 3.3 – Исходный код файла «./etc/listing.cpp»

```

46  matrices[0].render(); std::cout<<"\n"; matrices[1].render(); std::
    cout<<"\n";
47  //отправка данных
48  WSADATA hWSAData;
49  if(WSAStartup(0x0202,&hWSAData)!=0){
50      TheError("WSAStartup_failed");
51      return 1;
52  }
53  SOCKET hSocket = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);
54  struct sockaddr_in sockAddr;
55  sockAddr.sin_family = AF_INET; sockAddr.sin_port = htons(27015);
    sockAddr.sin_addr.S_un.S_addr = inet_addr("127.0.0.1");
56  if(sockAddr.sin_addr.S_un.S_addr==INADDR_NONE){
57      TheError("Outer_address_invalid");
58  }
59
60  if(connect(hSocket,(struct sockaddr*)&sockAddr,sizeof(sockAddr))
    ==-1){
61      TheError("Connection_failed");
62      return 1;
63  }
64  std::string data[2]; data[0] = matrices[0].serialize(); data[1] =
    matrices[1].serialize();
65  if(
66      send(hSocket,data[0].c_str(),(data[0].length()+1)*sizeof(TCHAR)
    ,0)==SOCKET_ERROR||
67      send(hSocket,data[1].c_str(),(data[1].length()+1)*sizeof(TCHAR)
    ,0)==SOCKET_ERROR){
68      TheError("Send_failed");
69      return 1;
70  }else{
71      shutdown(hSocket, SD_SEND);
72      std::cout<<"Data_has_been_sent_successfully:\n"<<data[0]<<"\n"
    <<data[1]<<"\n";
73  }
74  closesocket(hSocket);

```

Рисунок 3.4 – Исходный код файла «./etc/listing.cpp»

```
75  
76     system("pause");  
77     WSACleanup();  
78  
79     return 0;  
80 }
```

Рисунок 3.5 – Исходный код файла «./etc/listing.cpp»

### 3.3 Опциональный пакет «kpi.title.coursework»

Пакет позволяет сформировать титульные листы для курсовой работы. Как и любой title-пакет, определяет всего одну команду «maketitle». Для формирования требуется, чтобы были определены следующие команды метаданных:

- 1) docDepartment – название кафедры;
- 2) workSupervisor – руководитель работы, например: «Сокол В.Е.»;
- 3) workSupervisorPosition – статус руководителя работы, например «асистент каф. АСУ»;
- 4) docSubject – название курсовой работы;
- 5) studentGroup – группа автора;
- 6) studentName – ФИО автора;
- 7) docChecker – ФИО проверяющего;
- 8) docCheckerPosition – статус проверяющего, например «голова комісії професор».

Пример сформированных листов приведён в файле «./etc/title-coursework.pdf».