

Министерство образования и науки Российской Федерации
Иркутский национальный исследовательский технический университет

Институт кибернетики им. Е. И. Попова
Кафедра автоматизированных систем

Автоматизированное составление отчета на языке разметки \LaTeX

Отчет по лабораторной работе №
по дисциплине Информационный дизайн

Выполнил
Студент группы ИСТм-16-1

Принял

подпись

подпись

Костылев Д. А.

Фамилия И. О.

Григорьев С. В.

Фамилия И. О.

Иркутск 2016 г.

Содержание

1	Описание приложения, формирующего отчет	3
	Описание приложения, формирующего отчет	4
2	Описание языка разметки \LaTeX	5
	Описание языка разметки \LaTeX	6
3	Описание стилевых файлов	7
	Описание стилевых файлов	8
3.1	Листинги	8
3.2	Таблицы	9
	Список использованных источников	11
	Приложение А: Стиливые файлы	12
	Приложение Б: Скрипт, формирующий отчет	19

Введение

Выполнение лабораторной работы по дисциплине «Методы системного анализа» требовало разработку приложения для расчета оптимального плана перевозок. Для визуализации готового результата, а также для последовательного отображения выполняемых шагов был выбран формат отчета.

Основное приложение разрабатывалось на языке Python, версии 3, за удобство выполнения математических расчетов. В качестве инструмента верстки отчета был выбран язык разметки \LaTeX . Такое сочетание инструментов обеспечивает удобство интеграции двух изначально не интегрированных между собой областей.

1 Описание приложения, формирующего отчет

Для выполнения работы был выбран язык программирования Python 3.

По ходу выполнения программы, все данные, которые необходимы в отчете записывали в текстовый файл `solution.tex` встроенными методами. Пример приведен в листинге 1.1: строка 1 — открытие файла для записи, строка 2 — запись заголовка в файл.

Листинг 1.1— Пример записи текста в файл

```
1 report_output = open(REPORT_PATH, 'w')
2 report_write('\section{Решение}')
```

Также были созданы дополнительные функции для преобразования в выходной формат и записи в файл таблиц (1.2) и формул (1.3)

Листинг 1.2— Функция записи таблицы в выходной файл

```
1 def writetbl(t, caption="", debug=False):
2     if IS_OUTPUT or debug:
3         output = '\\begin{table}[H]\n'
4         output += '\\t\\centering\n'
5         output += '\\t\\normalsize\n'
6         output += '\\t\\caption{' + str(caption) + '}\n'
7         output += '\\t\\label{tbl:' + str(printed_table_no) + '}\n'
8         printed_table_no += 1
9         output += '\\t\\begin{tabular}{|} + 'c'|*len(t[0]) + '}\n'
10        output += '\\t\\t\\hline\n'
11
12        for row in t:
13            output += '\\t\\t'
14            for item in row:
15                output += str(item) + '&'
16            output = output[:-1] + '\\\\hline\n'
17
18        output += '\\n\\end{tabular}\n'
19        output += '\\n\\end{table}'
20        report_write(output, debug)
```

Листинг 1.3— Функция записи формулы в выходной файл

```
1 def writeeq(eq, num=False, debug=False):
2     if IS_OUTPUT or debug:
3         output = '\\vspace{-\\baselineskip}'
4         output += '\\begin{align}\n' if num else '\\begin{align*}\n'
5
6         output += eq
7
8         output += '\\n\\end{align}' if num else '\\n\\end{align*}'
9         report_write(output, debug)
```

Таким образом формирование формул будет выглядеть, как показано в листинге 1.4.

Листинг 1.4— Пример формирования формулы

```
1 restrictions = '\\begin{cases}'
2
3 for row, value in enumerate(tons):
4     for col, value in enumerate(value):
5         restrictions += str(value) + 'x_{' + str(row+1) + str(col+1) + '}'
6         restrictions += ' + '
7     restrictions = restrictions[:-1]
8     restrictions += '&\geqslant ' + str(min_transit[row])
9     restrictions += ' \\\\ \\n\\t\\t'
10
11 for row, value in enumerate(tons):
12     for col, value in enumerate(value):
13         restrictions += 'x_{' + str(col+1) + str(row+1) + '}' + ' + '
14     restrictions = restrictions[:-1]
15     restrictions += '&= ' + str(vehicles[row])
16     restrictions += ' \\\\ \\n\\t\\t'
17
18 restrictions += 'x_{ij} \geqslant 0, (i=1(1)'+ str(len(tons)) + '), (j=1(1)'+ str(len(tons[0])) + '),'
19 restrictions += '\\n\\t\\end{cases}'
20
21
22 writeeq(restrictions)
```

Перед завершением, приложение выполняет запускает компилятор (pdf latex) (листинг 1.5), после чего в текущей директории создается .pdf-файл отчета.

Листинг 1.5— Запуск компилятора

```
1 import os
2 cmd = 'pdf $\text{latex}$  main.tex'
3 os.system(cmd)
4 os.system(cmd) # second execution is for links
```

2 Описание языка разметки L^AT_EX

L^AT_EX— система верстки, ориентированная на производство научных математических документов высокого типографского качества. Система также вполне подходит для производства других видов документов, от простых писем до полностью сверстанных книг. L^AT_EXиспользует T_EXв качестве механизма верстки. [1]

К достоинствам выбранного языка разметки можно отнести способ формирования исходного кода, а именно отделение контента от его представления (шаблона, стилевого файла). На первом этапе подготавливается шаблон документа, а затем в ходе выполнения основного приложения в текстовый файл записываются необходимые данные, с соблюдением формата разметки, и запускается компилятор (`pdflatex`) и на выходе получается готовый .pdf-файл с отчетом о выполненной работе.

К недостаткам данного языка разметки можно отнести большой размер компилятора и сложность построения как стилевых файлов, так и разметки основного документа.

Началом L^AT_EX-документа является определение класса документа (например: `article`, `book`, `report` и др.)¹, который определяет некоторые начальные параметры, такие как уровни, заголовков, поля и др. Далее документ разделен на две части: преамбула и основной текст.

В преамбуле определены все параметры документа, переменные, окружения. Также существует возможность вынести все настройки из преамбулы в отдельный файл и подключить его либо как обычный текстовый файл командой `\input{ПАТН}`, либо как стилевой файл `\usepackage{ПАТН}`.

Для выполнения данной работы было создано два стилевых файла: с общими настройками и настройками отображения листингов (эти файлы подробно описаны в разделе 3).

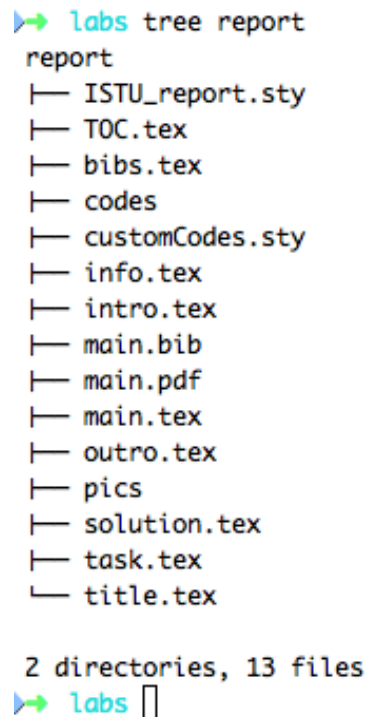
Основной текст документа заключен в окружение `document`.

Структура проекта представлена на рисунке 1. Каталоги `pics` и `codes` содержат изображения и листинги соответственно (подробнее, листинги описаны в разделе 3.1). `*.sty`-файлы содержат описание стилей. `main.bib` содержит описание всех использованных в документе ссылок на источники, а `bibs.tex` — описание оформления. `ТОС.tex` — описывает стиль оглавления.

Также отдельно вынесены переменные, используемые в документе, такие как имя студента и преподавателя, название дисциплины и др (файл `info.tex`).

Файл `solution.tex` формируется автоматически в ходе выполнения основного приложения (листинг Б.1).

¹ в файле отчета использовался сторонний класс `extarticle`



```
➤ labs tree report
report
├── ISTU_report.sty
├── TOC.tex
├── bibs.tex
├── codes
├── customCodes.sty
├── info.tex
├── intro.tex
├── main.bib
├── main.pdf
├── main.tex
├── outro.tex
├── pics
├── solution.tex
├── task.tex
└── title.tex

2 directories, 13 files
➤ labs
```

Рисунок 1 — Структура проекта

Основной текст может содержать в себе следующие элементы:

- простой текст;
- заголовки различных уровней;
- списки;
- листинги;
- ссылки, в том числе содержание, алфавитный указатель, список источников, указатель таблиц, формул, иллюстраций и др.;
- изображения и фигуры²;
- таблицы;
- формулы;
- окружения, определенные пользователем.

Стиль каждого из приведенных элементов может быть настроен отдельно.

²в данном случае, под фигурами понимаются иллюстрации, сформированными встроенным модулем TikZ или PGF

3 Описание стилевых файлов

В данном проекте все параметры документа вынесены в стилевые файлы, готовые к распространению[2]. Они имеют расширение*.sty. При составлении данных файлов были учтены следующие стандарты: стандарт оформления ИРНИТУ СТО 005-2015[3], ГОСТ 7.32—2001[4].

За основу взят класс extarticle, так как стандартные классы не поддерживают по умолчанию кегль основного текста более 12pt, согласно стандарту основной текст документа должен быть набран кеглем в 14pt. Также был указан формат листа А4. Подключены дополнительные пакеты для работы с кириллицей и русским языком (fontenc, inputenc, babel, pscyr). Основной кодировкой документа является utf-8.

Пакет `geometry` определяет поля страницы, согласно вышеуказанным стандартам, отступ от края страницы от границы текста должен быть:

- сверху — 1.5 см;
- справа — 1 см;
- снизу — 2 см;
- слева — 3 см.

Нумерация страниц указана по центру нижнего колонтитула.

Во всем документе, кроме листингов, использовалась стандартное семейство шрифтов Computer Modern. Гарнитура основного текста — Roman, заголовков — Bold Non-extended (начертание полужирное, капитель); моноширинная гарнитура Computer Modern Typewriter.

Основной текст имеет выключку по формату³. Отступ первой строки абзаца — 1,25см. Переносы запрещены. Интерлиньяж равен кеглю, это значит, что расстояние между базовыми линиями равно 14pt.

Гарнитура формул — Latin Modern Math, выключка по центру. Нумерация не использовалась.

Для иллюстраций используется верстка вразрез, при этом для каждого изображения выбирается оптимальный размер — информация, изображенная на рисунке должна быть легко читаема, но не занимать лишнее пространство на странице.

3.1 Листинги

Подключенный стилиевой файл позволяет включать листинги двумя способами: встроенный пакет `listings` и стороннее приложение-конвертер `Highlight t`(рисунок 2).

Преимущество первого метода состоит в том, что при его использовании не требуется предварительное преобразование. В данном

³англ: justified

проекте используется второй способ, поскольку данное приложение позволяет гибко настроить внешний вид листинга, но при этом автоматическое добавление листинга основного скрипта не использовалось, т. к. дополнительное преобразование усложнило бы процесс разработки.

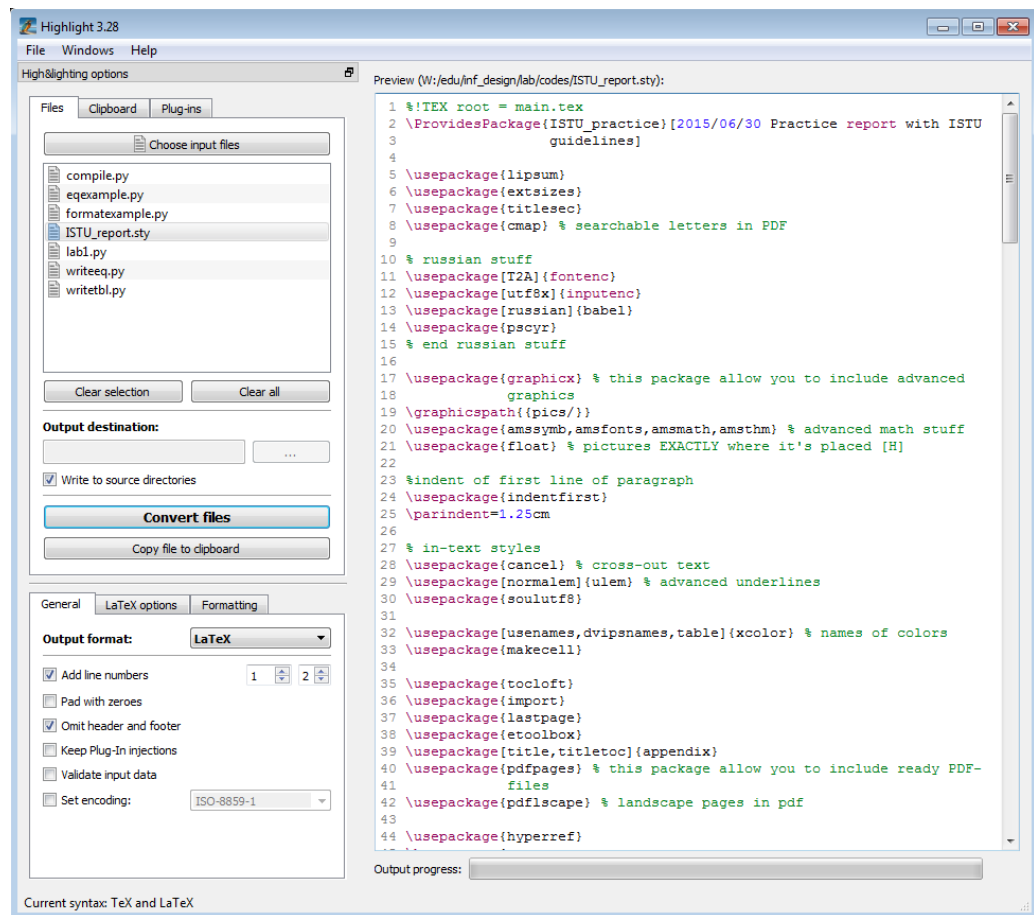


Рисунок 2 — Приложение highlight

Данное приложение преобразовывает исходный текст в язык разметки Latex, автоматически формируя разметку. Также на выходе создается стилевой файл, необходимый для корректного отображения листинга, но данном проекте он уже включен в стилевой файл `customCodes.sty` (листинг A.2). Также в результате выполнения экранируются служебные символы.

Гарнитура, используемая в листингах — свободная, моноширинная Inconsolata. Для сокращения используемого пространства расстояние между базовыми линиями (интерлиньяж) был выбран равный 0.68 от кегля основного текста. Кегль текста был задан как `\footnotesize`, что примерно равно 12pt.

3.2 Таблицы

Для отображения таблиц использовался пакет `longtables`, который позволяет дублировать заголовки таблиц и в начале каждой новой

страницы добавлять текст “продолжение таблицы...”, как того требуют стандарты.

Для удобства восприятия информации из таблиц, кегль всего текста внутри данного окружения был уменьшен примерно до 12pt, это позволило избежать переносов строк таблицы.

Также в стилевой файл включены такие пакеты, как: `multicol`, `multirow`, `booktabs` для более гибкой настройки таблиц (например, объединение ячеек, определяемый стиль границ), но в данном проекте не использовались.

Список использованных источников

- [1] Oetiker Tobias, Partl Hubert, Hyna Irene, Schlegl Elisabeth. The Not So Short Introduction to \LaTeX . — 2008. — URL: <http://tobi.oetiker.ch/lshort/lshort.pdf>.
- [2] Braams Johnnes. $\text{\LaTeX}2\epsilon$ for package writers // TUGboat 15.3. — 1999.
- [3] СТО 005-2015 Учебно-методическая деятельность. Оформление курсовых и дипломных проектов (работ) технических специальностей. — И., 2015.
- [4] ГОСТ 7.32—2001 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. — М. : Изд-во стандартов, 2001.

Приложение А: Стиливые файлы

Листинг А.1— Общий стилевой файл

```
1 %!TEX root = main.tex
2 \ProvidesPackage{ISTU_practice}[2015/06/30 Practice report with ISTU 2
3         guidelines]
4
5 \usepackage{lipsum}
6 \usepackage{extsizes}
7 \usepackage{titlesec}
8 \usepackage{cmap} % searchable letters in PDF
9
10 % russian stuff
11 \usepackage[T2A]{fontenc}
12 \usepackage[utf8x]{inputenc}
13 \usepackage[russian]{babel}
14 \usepackage{psycyr}
15 % end russian stuff
16
17 \usepackage{graphicx} % this package allow you to include advanced 2
18         graphics
19 \graphicspath{{pics/}}
20 \usepackage{amssymb,amsfonts,amsmath,amsthm} % advanced math stuff
21 \usepackage{float} % pictures EXACTLY where it's placed [H]
22
23 %indent of first line of paragraph
24 \usepackage{indentfirst}
25 \parindent=1.25cm
26
27 % in-text styles
28 \usepackage{cancel} % cross-out text
29 \usepackage[normalem]{ulem} % advanced underlines
30 \usepackage{soulutf8}
31
32 \usepackage[usenames,dvipsnames,table]{xcolor} % names of colors
33 \usepackage{makecell}
34
35 \usepackage{tocloft}
36 \usepackage{import}
37 \usepackage{lastpage}
38 \usepackage{etoolbox}
39 \usepackage[title,titletoc]{appendix}
40 \usepackage{pdfpages} % this package allow you to include ready PDF-2
41         files
42 \usepackage{pdflscape} % landscape pages in pdf
43
44 \usepackage{hyperref}
45 \hypersetup{
46     hidelinks
47 }
48
49 \usepackage{setspace}
50
51 % some stuff for anvaced tables
52 \usepackage{tabularx}
53 \usepackage{longtable}
54 \usepackage{hhline}
55 \usepackage{multirow}
```

```

56 % also "multicol" allow you to write part of text in two columns
57 \usepackage{multicol}
58 \usepackage{booktabs}
59 \usepackage{array}
60 \setlength{\multicolsep}{-7pt}
61
62 % restrict hyphenations...
63 \hyphenpenalty=10000
64 % ...and restrict lines go out of \textwidth (smth like justify)
65 \sloppy
66
67 % date format
68 \usepackage{datetime}
69 \newdateformat{onlyyear}{\THEYEAR~r.}
70
71 % margin
72 \usepackage{geometry}
73 \geometry{left=3cm}
74 \geometry{right=1cm}
75 \geometry{top=1.5cm}
76 \geometry{bottom=2cm}
77 \geometry{heightrounded}
78 \geometry{marginparwidth=2.5cm}
79 %\geometry{marginparsep=2cm}
80
81 \usepackage{marginnote}
82 \renewcommand*{\marginfont}{\color{red}\footnotesize}
83
84 \reversemarginpar
85
86 \renewcommand{\marginnote}[2][{}]{
87
88 \usepackage{afterpage}
89
90 %% style of unordered lists
91 \usepackage{enumitem}
92 \renewcommand{\labelitemi}{---}
93 \renewcommand{\labelenumi}{\asbuk{enumi})}
94 \renewcommand{\labelenumii}{\arabic{enumii})}
95 %% style of ordered lists
96 %% like 1.1.1 - third level
97 \renewcommand{\theenumi}{\arabic{enumi}}
98 \renewcommand{\labelenumi}{\arabic{enumi}.}
99 \renewcommand{\theenumii}{\arabic{enumii}}
100 \renewcommand{\labelenumii}{\arabic{enumi}.\arabic{enumii}.}
101 \renewcommand{\theenumiii}{\arabic{enumiii}}
102 \renewcommand{\labelenumiii}{\arabic{enumi}.\arabic{enumii}.\arabic{
103     enumiii}.}
104
105 \newenvironment{itemize*}
106   {\begin{itemize}[leftmargin=0mm, itemindent=2cm]
107     \vspace{-0.3cm}
108     \setlength{\itemsep}{1pt}
109     \setlength{\parskip}{1pt}}
110   {\end{itemize}}
111 \newenvironment{enumerate*}
112   {\begin{enumerate}[leftmargin=0mm, itemindent=2cm]
113     \vspace{-0.3cm}
114     \setlength{\itemsep}{1pt}

```

```

115     \setlength{\parskip}{1pt}}
116   {\end{enumerate}}
117 \newenvironment{arabicenumerate*}
118   {\begin{enumerate}[label={\arabic*}]}
119     \vspace{-0.3cm}
120     \setlength{\itemsep}{1pt}
121     \setlength{\parskip}{1pt}}
122   {\end{enumerate}}
123
124 % advanced headers'n'footers
125 \usepackage{fancyhdr}
126 \pagestyle{fancy}
127 \fancyhf{}
128 \fancyfoot[C]{\textcolor{gray}{0.4}}{\thepage}}
129 \fancyheadoffset{0mm}
130 \fancyfootoffset{0mm}
131 \renewcommand{\headrulewidth}{0pt}
132 \renewcommand{\footrulewidth}{0pt}
133 \fancypagestyle{plain}{
134   \fancyhf{}
135   \cfoot{\textcolor{gray}{0.4}}{\thepage}}
136 }
137
138 % headlines styles
139 % use section and subsection
140 \titleformat{\section}
141   {\bfseries\normalsize\centering}
142   {\thesection}
143   {1em}{}
144
145 \titleformat{\subsection}
146   {\normalsize\bfseries}
147   {\thesubsection}
148   {1em}{}
149
150 \titleformat{\subsubsection}
151   {\normalsize}
152   {\thesubsubsection}
153   {1em}{}
154
155 \titleformat{\chapter}
156   {\pagebreak\bfseries\normalsize\centering}
157   {\thechapter}
158   {1em}{}
159
160 \titlespacing*{\section}{\parindent}{*4}{*4}
161 \titlespacing*{\subsection}{\parindent}{*4}{*2}
162 \titlespacing*{\subsubsection}{\parindent}{*4}{*2}
163 \titlespacing*{\chapter}{\parindent}{*4}{*4}
164 % end headlines styles
165
166 % captions in subfigures
167 \numberwithin{table}{section}
168 \usepackage[tableposition=top]{caption}
169 \usepackage{subcaption}
170 %\numberwithin{figure}{section}
171
172 \renewcommand{\thesubfigure}{\asbuk{subfigure}}

```

```

173 \DeclareCaptionLabelFormat{gostfigure}{Рисунок #2}
174 \DeclareCaptionLabelFormat{gosttable}{Таблица #2}
175 \DeclareCaptionLabelSeparator{gost}{~---~}
176 \captionsetup{labelsep=gost}
177 \captionsetup[figure]{labelformat=gostfigure}
178 \captionsetup[table]{labelformat=gosttable}
179
180 \usepackage{framed} % allow you to put text im frame
181 \usepackage{marvosym} % symbols, smiles, icons
182
183 \usepackage{tocloft}
184 \renewcommand{\cfttoctitlefont}{\hspace{0.38\textwidth}}
185 \bfseries\normalsize}
186 \renewcommand{\cftsecfont}{\normalsize\hspace{1.25cm}}
187 \renewcommand{\cftsubsecfont}{\hspace{1.25cm}}
188 \renewcommand{\cftbeforesecskip}{0em}
189 \setcounter{tocdepth}{4}
190 \makeatletter
191 \renewcommand{\l@section}{\@dottedtocline{1}{1.25cm}{0.5cm}}
192 \renewcommand{\l@subsection}{\@dottedtocline{1}{1.75cm}{1cm}}
193 \renewcommand{\l@subsubsection}{\@dottedtocline{1}{2.25cm}{1.25cm}}
194 \makeatother
195
196 \bibliographystyle{ugost20081} %cite order
197 \addto\captionsrussian{\def\refname{}}
198
199 \makeatletter
200 \makeatletter
201 \renewenvironment{thebibliography}[1]
202     {\refname%
203     \thispagestyle{empty}
204     \@mkboth{\MakeUppercase\refname}{\MakeUppercase\refname}%
205     \list{\@biblabel{\@arabic\c@enumiv}}%
206         {\settowidth\labelwidth{\@biblabel{#1}}%
207         \leftmargin\labelwidth
208         \advance\leftmargin\labelsep
209         \@openbib@code
210         \usecounter{enumiv}%
211         \let\p@enumiv@empty
212         \renewcommand\theenumiv{\@arabic\c@enumiv}}%
213     \sloppy
214     \clubpenalty4000
215     \@clubpenalty \clubpenalty
216     \widowpenalty4000%
217     \sfcode'\.\@m}
218     {\def\@noitemerr
219     {\@latex@warning{Empty 'thebibliography' environment}}%
220     \endlist}
221 \makeatother
222
223 \newcounter{apps}
224 \setcounter{apps}{0}
225 \def\theapps{\Asbuk{apps}}
226 \newcommand{\app}[2]{
227     \def\thesection{\Asbuk{section}}
228     \refstepcounter{apps}
229     \refstepcounter{section}
230     \section*{Приложение~\theapps: #1}

```

```

231 \addcontentsline{toc}{section}{Приложение~\theapps: #1}
232 \label{sec:#2}
233 }
234
235 \usepackage{customCodes}
236
237 \newcommand{\undertext}[2]{\raisebox{-\baselineskip+0.17cm}{\par
238 \shortstack{\underline{#1}}\footnotesize{#2}}}}
239
240 \endinput

```

Листинг A.2— Стилиевой файл листингов

```

1 %!TEX root = main.tex
2 \ProvidesPackage{customCodes}[2013/01/13 Custom Codes]
3 % -----LISTINGS
4 % there are two ways to include listing:
5 % 1. "\usepackage{listings}"
6 %   troubles with russian symbols can occur
7 %   does not support some languages like CSS
8 %   styles are missing - under construction
9
10 % 2. stand-alone program "highlight": http://www.andre-simon.
11 de/doku/highlight/en/highlight.php
12
13 % 1.
14 \usepackage{listings}
15 \lstset{ %
16     backgroundcolor=\color{gray}{0.5}, % choose the background color;
17                                     you must add \usepackage{color} or \usepackage{xcolor}
18     xcolor}
19     inputencoding=utf8x,
20     keepspaces = true
21     extendedchars=false,
22     basicstyle=\footnotesize, % the size of the fonts that are
23                     used for the code
24     breakatwhitespace=false, % sets if automatic breaks should
25                             only happen at whitespace
26     breaklines=true, % sets automatic line breaking
27     captionpos=t, % sets the caption-position to
28                     bottom
29     commentstyle=\color{green}, % comment style
30     deletekeywords={...}, % if you want to delete keywords
31                     from the given language
32     escapeinside={\%*}{*}, % if you want to add LaTeX within
33                     your code
34 %     extendedchars=true, % lets you use non-ASCII
35 %                         characters; for 8-bits encodings only, does not work
36 %                         with UTF-8
37     frame=single, % adds a frame around the code
38     keepspaces=true, % keeps spaces in text, useful
39                     for keeping indentation of code (possibly needs
40                     columns=flexible)
41     keywordstyle=\color{blue}, % keyword style
42 %     language=Python, % the language of the code
43     otherkeywords={*,...}, % if you want to add more
44                     keywords to the set
45     numbers=left, % where to put the line-numbers;

```



```

46         possible values are (none, left, right)
47     numbersep=5pt,                % how far the line-numbers are 2
48         from the code
49     numberstyle=\tiny\color{gray}, % the style that is used for the 2
50         line-numbers
51     rulecolor=\color{black},       % if not set, the frame-color may 2
52         be changed on line-breaks within not-black text (e.g. 2
53         comments (green here))
54     showspaces=false,             % show spaces everywhere adding 2
55         particular underscores; it overrides 2
56         'showstringspaces'
57     showstringspaces=false,       % underline spaces within strings 2
58         only
59     showtabs=false,              % show tabs within strings adding 2
60         particular underscores
61     stepnumber=2,                % the step between two line-2
62         numbers. If it's 1, each line will be numbered
63     stringstyle=\color{red},      % string literal style
64     tabsize=2,                  % sets default tabsize to 2 2
65         spaces
66     title=\lstname               % show the filename of files 2
67         included with \lstinputlisting; also try caption instead of 2
68         title
69 }
70 \renewcommand{\lstlistingname}{Листинг}
71
72 % Styles:
73 \lstdefinestyle{customc}{
74     belowcaptionskip=1\baselineskip,
75     breaklines=true,
76     frame=L,
77     xleftmargin=\parindent,
78     language=Python,
79 %     showstringspaces=false,
80     basicstyle=\footnotesize\ttfamily,
81     keywordstyle=\bfseries\color{green!40!black},
82     commentstyle=\itshape\color{purple!40!black},
83     identifierstyle=\color{blue},
84     stringstyle=\color{orange},
85 }
86 %     end 1.
87
88 %     2.
89 \definecolor{lstcodeBG}{rgb}{0.95,0.95,0.95}
90 \definecolor{lstcodeFR}{rgb}{0.95,0.95,0.95}
91
92 \newcounter{lstcodeN}[section]
93 \setcounter{lstcodeN}{2}
94 \def\thelstcodeN{\thesection.\arabic{lstcodeN}}
95
96 \newcommand{\lstcode}[5]{
97     \refstepcounter{lstcodeN}\label{lst:#5}
98     \vspace{0.2cm}
99     Листинг \thelstcodeN --- #4
100     \vspace{-0.2cm}
101     #3
102     \setstretch{#2}
103     \begin{framed}
104         \input{#1}
105     \end{framed}
106     \normalsize

```

```

107     \singlespacing
108 }
109
110 % escape characters DO NOT EDIT
111 \newsavebox{\hlboxclosebrace}%
112 \newsavebox{\hlboxopenbrace}%
113 \newsavebox{\hlboxbackslash}%
114 \newsavebox{\hlboxlessthan}%
115 \newsavebox{\hlboxgreaterthan}%
116 \newsavebox{\hlboxdollar}%
117 \newsavebox{\hlboxunderscore}%
118 \newsavebox{\hlboxand}%
119 \newsavebox{\hlboxhash}%
120 \newsavebox{\hlboxat}%
121 \newsavebox{\hlboxpercent}%
122 \newsavebox{\hlboxhat}%
123 \newsavebox{\hlboxsinglequote}%
124 \newsavebox{\hlboxbacktick}%
125
126 \setbox\hlboxopenbrace=\hbox{\verb{.}%
127 \setbox\hlboxclosebrace=\hbox{\verb{.}%
128 \setbox\hlboxlessthan=\hbox{\verb{<}%
129 \setbox\hlboxdollar=\hbox{\verb{$.}%
130 \setbox\hlboxunderscore=\hbox{\verb{_.}%
131 \setbox\hlboxand=\hbox{\verb{&}.}%
132 \setbox\hlboxhash=\hbox{\verb{#.}%
133 \setbox\hlboxat=\hbox{\verb{@.}%
134 \setbox\hlboxbackslash=\hbox{\verb{\.}%
135 \setbox\hlboxgreaterthan=\hbox{\verb{>}.}%
136 \setbox\hlboxpercent=\hbox{\verb{%.}%
137 \setbox\hlboxhat=\hbox{\verb{^.}%
138 \setbox\hlboxsinglequote=\hbox{\verb{'}.}%
139 \setbox\hlboxbacktick=\hbox{\verb{`.}%
140 \setbox\hlboxhat=\hbox{\verb{^.}%
141
142 % XCODE preinstalled theme for highlight
143 \newcommand{\hlstd}[1]{\textcolor{rgb}{0,0,0}{\#1}}
144 \newcommand{\hlnum}[1]{\textcolor{rgb}{0,0,0}{\#1}}
145 \newcommand{\hlesc}[1]{\textcolor{rgb}{0.64,0.08,0.08}{\#1}}
146 \newcommand{\hlstr}[1]{\textcolor{rgb}{0.64,0.08,0.08}{\#1}}
147 \newcommand{\hlpps}[1]{\textcolor{rgb}{0,0,1}{\#1}}
148 \newcommand{\hlslc}[1]{\textcolor{rgb}{0,0.5,0}{\#1}}
149 \newcommand{\hlcom}[1]{\textcolor{rgb}{0,0.5,0}{\#1}}
150 \newcommand{\hlppc}[1]{\textcolor{rgb}{0,0,1}{\#1}}
151 \newcommand{\hlopt}[1]{\textcolor{rgb}{0,0,0}{\#1}}
152 \newcommand{\hlipl}[1]{\textcolor{rgb}{0.64,0.08,0.08}{\#1}}
153 \newcommand{\hlilin}[1]{\textcolor{rgb}{0.17,0.57,0.69}{\#1}}
154 \newcommand{\hlkwa}[1]{\textcolor{rgb}{0,0,1}{\#1}}
155 \newcommand{\hlkwb}[1]{\textcolor{rgb}{0,0,1}{\#1}}
156 \newcommand{\hlkwc}[1]{\textcolor{rgb}{0.17,0.57,0.69}{\#1}}
157 \newcommand{\hlkwd}[1]{\textcolor{rgb}{0,0,0}{\#1}}
158 \definecolor{bgcolor}{gray}{0.95}
159 % end 2.
160
161 \endinput

```

Приложение Б: Скрипт, формирующий отчет

Листинг Б.1— Исходный код приложения

```
1 #####
2 from IPython.display import display, Latex, Math, Markdown
3 import numpy as np
4 np.seterr(divide='ignore', invalid='ignore')
5 import pandas as pd
6 import math
7 import copy
8 from ipy_table import make_table, set_row_style
9 import re
10 import random
11 #####
12
13 ROUND_ACCURACY = 4
14 MAX_ITERATIONS = 11
15 REPORT_PATH = 'report/solution.tex'
16
17 #####
18
19 IS_OUTPUT = True
20
21 report_output = open(REPORT_PATH, 'w')
22 def report_write(s, debug=False):
23     if IS_OUTPUT or debug:
24         report_output.write(s.replace('&nbsp;', ' ')+'\n\n')
25
26 def writetbl(t, caption="", debug=False):
27     if IS_OUTPUT or debug:
28         output = '\\begin{table}[H]\n'
29         output += '\\t\\centering\n'
30         output += '\\t\\normalsize\n'
31         output += '\\t\\caption{' + str(caption) + '}\n'
32         output += '\\t\\label{tbl:' + str(printed_table_no) + '}\n'
33         # printed_table_no += 1
34         output += '\\t\\begin{tabular}{|' + 'c|'*len(t[0]) + '}\n'
35         output += '\\t\\t\\hline\n'
36
37         for row in t:
38             output += '\\t\\t'
39             for item in row:
40                 output += str(item) + '&'
41             output = output[:-1] + '\\t\\t\\hline\n'
42
43         output += '\\n\\end{tabular}\n'
44         output += '\\n\\end{table}'
45         report_write(output, debug)
46
47 def writeeq(eq, num=False, debug=False):
48     if IS_OUTPUT or debug:
49         output = '\\vspace{-\\baselineskip}'
50         output += '\\begin{align}\n' if num else '\\begin{align*}\n'
51
52         output += eq
53
54         output += '\\n\\end{align}' if num else '\\n\\end{align*}'
55         report_write(output, debug)
56 #####
57
```

```

58 printed_table_no = 1
59
60 #####
61
62 tons = [
63     [25,20,50],
64     [15,10,0],
65     [10,40,8],
66 ]
67 cost = [
68     [15,10,30],
69     [25, 6, 0],
70     [30, 5,10],
71 ]
72 min_transit = [
73     500,
74     200,
75     100
76 ]
77 vehicles = [55,95,30]
78
79 # test set source data
80 # tons = [
81 #     [25,20,50,50],
82 #     [20,12, 0,45],
83 #     [15,10, 0,40],
84 #     [10,40, 8,25],
85 # ]
86 # cost = [
87 #     [15,10,30,25],
88 #     [20, 8, 0,30],
89 #     [25, 6, 0,15],
90 #     [30, 5,10,45],
91 # ]
92 # min_transit = [
93 #     500,
94 #     200,
95 #     200,
96 #     100
97 # ]
98 # vehicles = [55,95,30,45]
99
100 table_1 = []
101 table_1.append([])
102 table_1[0].append('№ линии\\textbackslash № судна')
103 for i in range(len(vehicles)):
104     table_1[0].append(i+1)
105 table_1[0].append('Минимальный объем 2
106                     перевозок')
107
108 for index,value in enumerate(tons):
109     table_1.append([])
110     table_1[index+1].append(index+1)
111     for idx,item in enumerate(value):
112         table_1[index+1].append(str(item)+'/'+str(cost[index][idx]))
113     table_1[index+1].append(min_transit[index])
114
115 table_1.append([])
116 table_1[-1].append('Кол-во кораблей')
117
118 for item in vehicles:
119     table_1[-1].append(item)

```

```

120 table_1[-1].append("")
121
122 writetbl(table_1, 'Исходные данные')
123
124 #####
125 report_write('\\section{Математическая модель}')
126
127 report_write("")
128 Обозначим через  $x_{ij}$  --- количество
129 судов, перевозимого
130 по линиям перевозки,
131  $c_{ij}$  --- стоимость
132 перевозки.
133
134 Целевая функция ---  $\min F(x_{ij}) = \min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$ 
135
136
137
138 Целевая функция отражает минимальные
139 транспортные издержки, при которых
140 запросы всех потребителей
141 удовлетворены.
142
143 Требуется определить множество
144 переменных  $x_{ij}$   $\geq 0$ ,
145 удовлетворяющих следующим условиям:
146
147  $\sum_{j=1}^n x_{ij} \leq a_i$ , где  $i = 1, 2, \dots, m$ 
148
149
150  $\sum_{i=1}^m x_{ij} = N_j$ , где  $j = 1, 2, \dots, n$ 
151
152
153 В ограничениях  $a_i$  --- минимальный
154 объем перевозок на линии  $i$ ;  $N_j$  ---
155 количество судов вида  $j$ .
156
157 \\subsection{Индивидуальное задание}
158
159 Обозначим через  $x_{ij}$  число судов типа
160  $j$  ( $j = 1, 2, 3$ ), которое
161 планируется
162 закрепить за
163 регулярной линией  $i$ 
164 ( $i = 1, 2, 3$ ).
165
166 С учетом введенных обозначений
167 математическая модель задачи:
168
169 Целевая функция:
170 ""
171
172 math_model = '\\min Z_{x_{ij}} = \\min_{x_{ij}} \\left( \\right.'
173 for row,value in enumerate(cost):
174     for col,value in enumerate(value):
175         math_model += str(value)+'x_{'+str(row+1)+str(col+1)+'}+'
176     math_model += '\\\\n\\t'
177 math_model = math_model[:-6] + '\\left. \\right)'
178
179 writeeq(math_model)
180

```

```

181 report_write(', при ограничениях:')
182
183 restrictions = '\\begin{cases}'
184
185 for row, value in enumerate(tons):
186     for col, value in enumerate(value):
187         restrictions += str(value) + 'x_{ ' + str(row+1) + str(col+1) + ' } + '
188     restrictions = restrictions[:-1]
189     restrictions += '&\\geqslant ' + str(min_transit[row])
190     restrictions += ' \\\\ \\n\\t\\t'
191
192
193 for row, value in enumerate(tons):
194     for col, value in enumerate(value):
195         restrictions += 'x_{ ' + str(col+1) + str(row+1) + ' } + '
196     restrictions = restrictions[:-1]
197     restrictions += '&= ' + str(vehicles[row])
198     restrictions += ' \\\\ \\n\\t\\t'
199
200 restrictions += 'x_{ij} \\geqslant 0, (i=1(1)' + str(len(tons)) + '), (j=1(1)' + str(len(tons[0])) + '),'
201 restrictions += '\\n\\t\\end{cases}'
202
203
204 writeeq(restrictions)
205
206 report_write('
207 Обратим внимание, что в целевой
208 функции коэффициент при переменной
209 равен 1000,
210 что значительно больше любого из
211 остальных коэффициентов целевой
212 функции.
213 Тем самым использование судов
214 третьего типа на второй регулярной
215 линии "заблокировано",
216 так как при $x_{32} \\neq 0$ значение целевой
217 функции резко
218 возрастает, и алгоритм,
219 сформированный на основе метода
220 симплекс-таблиц, выведет переменную $x_{32}$ из числа базисных переменных,
221 т.е. определит значение равным нулю.
222
223 В системе ограничений вида неравенств
224 коэффициент при равен нулю. Тем самым,
225 отражено то,
226 что на судах 3-го типа по 2-ой
227 регулярной линии количество
228 перевозимого груза может быть только
229 равным нулю.
230
231 ')
232
233 #####
234 report_write('\\section{Решение}')
235 report_write('\\subsection{Нахождение начального
236 допустимого базисного решения.
237 Метод Данцига.}')
238
239 report_write('Приведем индексы матрицы
240 ограничений к такому виду:')
241
242 wided_table_cols = len(tons)*len(tons[0])+len(tons)+1

```

```

243 wided_table_rows = 2*len(tons)
244 wided_table = [[0]*wided_table_cols for i in range(wided_table_rows)]
245 wided_table_visible = [[0]*(wided_table_cols) for i in range(2
246                     wided_table_rows+1)]
247
248 # header of the table
249 for i in range(wided_table_cols):
250     wided_table_visible[0][i] = '$x_{' + str(i) + '}$'
251
252 # first col (restrictions) A_0
253 for i in range(wided_table_rows):
254     try:
255         wided_table[i][0] = min_transit[i]
256         wided_table_visible[i+1][0] = '$' + str(min_transit[i]) + '$'
257     except IndexError:
258         wided_table[i][0] = vehicles[i-len(min_transit)]
259         wided_table_visible[i+1][0] = '$' + str(vehicles[i-len(2
260                     min_transit)]) + '$'
261
262 # coeffs of restrictions
263 for row, value in enumerate(tons):
264     for col, item in enumerate(value):
265         curr_col = len(tons)*col+row+1
266         wided_table[row][curr_col] = item
267         wided_table_visible[row+1][curr_col] = '$\\mathbf{' + str(item)2
268                     + '}$'
269
270 # additional -1 to make equations with 0
271 for item in range(len(tons)):
272     wided_table[item][len(tons)**2+item+1] = -1
273     wided_table_visible[item+1][len(tons)**2+item+1] = '$\\mathbf{' + 2
274                     str(-1) + '}$'
275
276 # coeffs of vehicles restrictions
277 for row, value in enumerate(tons):
278     for col, item in enumerate(value):
279         curr_col = len(tons)*col+row+1
280         wided_table[row+len(tons)][curr_col] = 1
281         wided_table_visible[row+1+len(tons)][curr_col] = '$\\mathbf{' + 2
282                     str(1) + '}$'
283
284 writetbl(wided_table_visible)
285 # writetbl(wided_table)
286
287
288 #####
289
290 basis_idx = []
291 basis_val = []
292 text = 'Вводим в базис произвольные 2
293     переменные: '
294
295 for i in range(wided_table_rows):
296     basis_idx.append(i)
297     basis_val.append([ row[i] for row in wided_table ])
298     text += '$A_{' + str(i) + '}$, '
299
300 not_basis_idx = sorted(list(set(range(wided_table_cols)) - set(2
301                     basis_idx)))
302 not_basis_val = []
303 for idx in not_basis_idx:
304     not_basis_val.append([ row[idx] for row in wided_table ])

```

```

305 text = text[:-2] + ' .'
306 report_write(text)
307
308 #####
309
310 aux_array = [random.randrange(0,10) for i in range(len(basis_idx))]
311
312 SUCCESS = False
313 iteration = 1
314 while iteration < MAX_ITERATIONS + 1:
315     report_write('\subsubsection{Мар ' + str(iteration) + '}')
316
317     # equations on step
318     report_write('\small')
319     for idx, nbas in enumerate(not_basis_idx):
320         text = ""
321         for i in range(len(basis_val[0])):
322             text += '\tA_{'+str(nbas)+'} &= ' + str(not_basis_val[2
323                 not_basis_idx.index(nbas)][i]) + ' = '
324             for bas in basis_idx:
325                 text += str(basis_val[basis_idx.index(bas)][i]) + 'x_{2
326                     ' + str(bas) + '-' + str(nbas) + '}' + '
327             text = str(text[:-3]) + ' \\\ \n' # remove extra plus on 2
328                 the end
329             text = str(text[:-5])
330             writeeq(text)
331         report_write('\normalsize')
332
333     #####
334
335     basis_solutions = []
336     basis_val_transposed = list(np.array(basis_val).transpose())
337     for item in not_basis_val:
338         basis_solutions.append(
339             list(np.linalg.solve(basis_val_transposed, item))
340         )
341
342     # rounding result
343     # dirty hack with float(str())
344     basis_solutions = [[float(str(round(item,ROUND_ACCURACY))) for 2
345         item in sol] for sol in basis_solutions]
346
347
348     report_write('Решения уравнений:')
349
350     basis_solutions_visible = [['$x_{'+str(i)+'}$' for i in 2
351         basis_idx]]
352     for nbas_idx,nbas in enumerate(not_basis_idx):
353         text = '\begin{array}{'+c'*len(basis_solutions[0])+'}\n\t'
354         for bas_idx,bas in enumerate(basis_idx):
355             text += 'x_{'+str(bas)+'-' + str(nbas) + '} = '\
356                 + str(basis_solutions[nbas_idx][bas_idx]) + ';\n\t'
357             text += '\\\ \n\t' if bas_idx != 0 and bas_idx%3 == 0 2
358                 else ""
359             basis_solutions_visible.append(['$A_{'+str(nbas)+'}$'+2
360                 basis_solutions[nbas_idx])
361         text += '\n\end{array}'
362         # writeeq(text)
363
364     writetbl(basis_solutions_visible,caption='Решения 2
365         уравнений в виде таблицы')
366

```



```

367 TO_COMPARE = 0
368 report_write('Сравнивая решения при $x_' + str(2
369             TO_COMPARE) + '$:')
370
371 text = '\t'
372 min_solution = 999999
373 min_idx = 0
374 for sol_idx, sol in enumerate(basis_solutions):
375     text += 'x_{0}'+str(not_basis_idx[sol_idx])+'} = ' + str(sol[2
376             TO_COMPARE]) + ';\n'
377     text = text[:-3] + '\\\\n\t' if sol_idx != 0 and sol_idx%3 2
378         == 0 else text
379     if sol[TO_COMPARE] > 0:
380         if sol[TO_COMPARE] < min_solution:
381             min_solution = sol[TO_COMPARE]
382             min_idx = sol_idx
383
384 text = text[:-3]
385 writeeq(text)
386
387 if min_solution != 999999:
388     report_write('Минимальный элемент: $x_{'+\
389                 str(TO_COMPARE) + '-'+\
390                 str(not_basis_idx[min_idx])+'} = ' +\
391                 str(min_solution) + '$.')
392
393 exclude_eq_idx = -1
394 theta0 = -1
395 for sol_idx, sol in enumerate(basis_solutions[min_idx]):
396     # if sol_idx != 0:
397     if sol > 0:
398         if aux_array[sol_idx]/sol > 0:
399             exclude_eq_idx = sol_idx
400             theta0 = round(aux_array[sol_idx]/sol, ROUND_ACCURACY)
401             break
402
403 # if there's no positive coeffs, it means that we find optimal 2
404 basis solution
405 if (exclude_eq_idx == -1 and theta0 == -1) or min_solution == 2
406     999999:
407     report_write('Нет положительных 2
408                 коэффицентов, 2
409                 соответственно допустимое 2
410                 базисное решение:')
411
412 text = ""
413
414 for sol_idx, sol in enumerate(basis_solutions[min_idx]):
415     text += 'x_{'+str(not_basis_idx[min_idx])+'-'+str(2
416             basis_idx[sol_idx])+'}^{\\*} = '\
417     + str(sol) + ',\n'
418     text = text[:-3] + '\\\\n\t' if sol_idx != 0 and 2
419         sol_idx%3 == 0 else text
420 text = text[:-3]
421 writeeq(text)
422 SUCCESS = True
423 break
424
425 report_write('Вводим в базис вектор $A_{' + str(2
426             not_basis_idx[min_idx]) + '} $ и запишем 2
427             для него уравнение:')
428

```

```

429 report_write('\\small')
430 text = ""
431 text += 'A_{'+str(not_basis_idx[min_idx])+'} = '
432 for sol_idx,sol in enumerate(basis_solutions[min_idx]):
433     if sol_idx != 0:
434         if sol > 0:
435             text += '+'
436             text += str(sol) +\
437                 'x_{'+str(not_basis_idx[min_idx])+'}-'+str(basis_idx[sol_idx])+')
438             '}'
439 writeeq(text)
440 report_write('\\normalsize')
441
442 #####
443
444 if iteration == 1:
445     report_write('Введем вспомогательный \
446                 вектор со случайными \
447                 значениями:')
448 else:
449     report_write('Вспомогательный вектор на \
450                 этом шаге:')
451
452 text = ""
453 for idx in range(len(aux_array)):
454     text += '\\rho_{ ' if idx == 0 else '\\omega_{ '
455     text += str(idx)+' } '
456     text += ' = '+str(aux_array[idx]) + ';\\ '
457 text = text[:-3]
458 writeeq(text)
459
460 # including new vector
461 basis_idx.append(not_basis_idx[min_idx])
462 basis_val.append(not_basis_val[min_idx])
463 not_basis_idx.append(basis_idx[exclude_eq_idx])
464 not_basis_val.append(basis_val[exclude_eq_idx])
465 del not_basis_idx[min_idx]
466 del not_basis_val[min_idx]
467 del basis_idx[exclude_eq_idx]
468 del basis_val[exclude_eq_idx]
469
470 # sorting like in wided table
471 basis_idx, basis_val = [list(i) for i in zip(*sorted(zip(basis_idx,\
472                                                         basis_val)))]
473 not_basis_idx, not_basis_val = [list(i) for i in zip(*sorted(zip(\
474                                                         not_basis_idx, not_basis_val)))]
475
476 text = 'Выводим из базиса вектор $A_{'+\
477       str(basis_idx[exclude_eq_idx]) +\
478       '$, т.к. $'+\
479       '\\theta_0 = \\frac{' +\
480       str(aux_array[exclude_eq_idx])+'} {' +\
481       str(basis_solutions[min_idx][exclude_eq_idx])+'} $ = '\
482       + str(theta0)
483
484 for idx,item in enumerate(aux_array):
485     aux_array[idx] = round(item + theta0*basis_solutions[min_idx][\
486                             idx],ROUND_ACCURACY)
487
488 report_write(text)
489

```

```

490     iteration += 1
491
492 if not SUCCESS:
493     report_write('Не найдено решения за ' + \
494                 str(MAX_ITERATIONS) + \
495                 ' шагов. Или что-то пошло не так или \
496                 надо увеличить число шагов.')
497     report_output.close()
498     exit()
499
500 report_write('\\subsection{Переход от начального \
501             допустимого решения к первой \
502             симплекс-таблице}')
503
504 report_write('Разложим небазисные векторы \
505             по найденному методом Данцига \
506             базису:')
507
508 simplex_tbl = []
509 conversion_tbl = []
510
511 report_write('\\small')
512 for nbas_idx, nbas in enumerate(not_basis_val):
513     text = ""
514     for item_idx, item in enumerate(nbas):
515         text += str(nbas[item_idx]) + ' &= '
516         for bas_idx, bas in enumerate(basis_idx):
517             if bas_idx != 0:
518                 if not basis_val[bas_idx][item_idx] < 0:
519                     text += '+'
520                 text += str(basis_val[bas_idx][item_idx]) + \
521                     'x_{ ' + str(bas) + '-' + str(not_basis_idx[nbas_idx]) + ' }'
522             ,
523
524     text += ' \\\\ \\n\\t'
525     text = text[:-3]
526     writeeq(text)
527
528 report_write('\\normalsize')
529
530 report_write('Решая каждую из систем \
531             уравнений, получим:')
532 report_write('\\footnotesize')
533
534 for item in not_basis_val:
535     conversion_tbl.append(
536         list(np.linalg.solve(basis_val_transposed, item))
537     )
538 conversion_tbl = [[float(str(round(j, ROUND_ACCURACY))) for j in i] for \
539                   i in conversion_tbl]
540
541 text = '\\begin{array}{ ' + 'c'*len(conversion_tbl[0]) + ' }\\n'
542 for row_idx, row in enumerate(conversion_tbl):
543     text += '\\t'
544     for item_idx, item in enumerate(row):
545         text += 'x_{ ' + \
546             str(basis_idx[item_idx]) + \
547             '-' + \
548             str(not_basis_idx[row_idx]) + ' } = ' \
549             + str(item) + ', & '
550     text = text[:-4] + ' \\\\ \\n'
551

```

```

552 text = text[:-4] + '\n\\end{array}'
553 writeeq(text)
554 report_write('\\normalsize')
555 #####
556
557 report_write('\\subsection{Решение методом }
558             полного исключения Гаусса}')
559
560 # flatten cost array and append zeros to simplex table length
561 # (-1 coz first item is empty)
562 cost_flat = list(np.array(cost).flatten())
563 cost_flat = [""] + cost_flat + [0]*((len(basis_idx)+len(not_basis_idx))
564                                     -len(cost_flat)-1)
565
566 ####
567 # Fill simplex table with values from conversion table if x is not in
568 basis
569 # if x in basis so 1 at cross, 0 others
570 ####
571 curr_sol = 0
572 for idx in range(len(wided_table[0])):
573     if idx in basis_idx:
574         simplex_tbl.append(
575             [1 if i == basis_idx.index(idx) else 0 for i in range(len(
576                 basis_idx))] + [0]
577         )
578     else:
579         simplex_tbl.append(conversion_tbl[curr_sol] + [cost_flat[idx]])
580         curr_sol += 1
581
582 basis_left_col = []
583 for item in basis_idx:
584     basis_left_col.append(float(cost_flat[item]))
585
586 simplex_tbl = list(np.array(simplex_tbl).transpose())
587 simplex_tbl = [list(item) for item in simplex_tbl]
588
589 simplex_tbl = [cost_flat] + simplex_tbl
590
591 for row_idx, row in enumerate(simplex_tbl):
592     for item_idx, item in enumerate(row):
593         simplex_tbl[row_idx][item_idx] = float(item) if item != ""
594         else ""
595
596 init_target_func = 0
597 for item_idx, item in enumerate(simplex_tbl):
598     if item_idx != 0 and item_idx != len(simplex_tbl)-1:
599         init_target_func += float(item[0])*float(basis_left_col[
600             item_idx-1])
601
602
603 simplex_tbl[len(simplex_tbl)-1][0] = init_target_func
604
605 direction_col = -1
606 direction_row = -1
607
608 # havePositive = False
609 SUCCESS = False
610 iteration = 1
611 while iteration < MAX_ITERATIONS + 1:
612     #####
613     # find direction col and row

```

```

614 min_col_val = float('NaN')
615 for item_idx, item in enumerate(simplex_tbl[-1]):
616     if item_idx != 0:
617         if float(item) > 0:
618             if not item < min_col_val:
619                 min_col_val = item
620                 direction_col = item_idx
621
622 if math.isnan(min_col_val):
623     direction_row = 0
624     direction_col = 0
625
626 if direction_col != 0:
627     min_row_val = float('inf')
628     for item_idx, item in enumerate(simplex_tbl):
629         if item_idx != 0 and item_idx != len(simplex_tbl)-1:
630             if simplex_tbl[item_idx][direction_col] > 0.0:
631
632                 divided = simplex_tbl[item_idx][0] / simplex_tbl[2]
633                     item_idx][direction_col]
634
635                 if divided < min_row_val:
636                     min_row_val = divided
637                     direction_row = item_idx
638
639 if direction_col != 0:
640     havePositive = False
641     for row in simplex_tbl:
642         if row[direction_col] > 0:
643             havePositive = True
644             break
645
646 #####
647 # make visual representation of simplex table
648 simplex_tbl_visible = []
649
650 # first row
651 simplex_tbl_visible.append(['$c$', ""])
652 simplex_tbl_visible[0] += ['$' + str(item) + '$' if item != " "
653                             else " for item in simplex_tbl[0]]
654
655 # second row
656 simplex_tbl_visible.append(["", '$B_x$', '$a_{i0}$'])
657 for i in range(len(simplex_tbl[0])-1):
658     if i == direction_col-1 and direction_col != 0:
659         simplex_tbl_visible[1].append('$\\mathbf{A_{'+str(i+1)+'}}$')
660         '$')
661     else:
662         simplex_tbl_visible[1].append('$A_{'+str(i+1)+'}$')
663
664 # body of table
665 for idx in range(len(simplex_tbl)):
666     if idx != 0:
667         if idx == len(simplex_tbl)-1:
668             simplex_tbl_visible.append(
669                 ["", '$\\Delta$'] + \
670                 ['$' + str(item) + '$' for item in simplex_tbl[idx]]
671                 )
672         else:
673             if idx == direction_row and direction_row != 0:
674                 simplex_tbl_visible.append(

```

```

676         ['$' + str(basis_left_col[idx-1]) + '$' ,
677         '$\\mathbf{x}_{' + str(basis_idx[idx-1]) + '}}$']
678     +\
679     ['$\\mathbf{' + str(item) + '}$' if item_idx >
680     == direction_col else '$' + str(
681     item) + '$' for item_idx,item in
682     enumerate(simplex_tbl[idx])]
683     )
684     else:
685         simplex_tbl_visible.append(
686         [basis_left_col[idx-1] , '$x_{' + str(basis_idx[
687         idx-1]) + '}}$'] +
688         \
689         ['$' + str(item) + '$' for item in simplex_tbl[
690         idx]]
691         )
692
693     report_write('\\begin{landscape}')
694     report_write('\\subsubsection{Шаг ' + str(iteration) + '}')
695
696     writetbl(simplex_tbl_visible,debug=True)
697
698     if direction_col != 0 and direction_row != 0:
699         report_write('Направляющий столбец: ' + str(
700         direction_col))
701         report_write('Направляющая строка: ' + str(
702         direction_row))
703         report_write('Разрешающий элемент: ' + str(
704         simplex_tbl[direction_row][direction_col]))
705
706
707     report_write('\\end{landscape}')
708
709     if not havePositive:
710         SUCCESS = False
711         break
712
713     if math.isnan(min_col_val):
714         target_func = 0
715         for idx in range(len(simplex_tbl)-1):
716             if idx != 0:
717                 target_func += simplex_tbl[idx][0]*basis_left_col[idx-
718                 1]
719             simplex_tbl[row_idx][item_idx] = target_func
720             SUCCESS = True
721             break
722
723     new_simplex_tbl = copy.deepcopy(simplex_tbl)
724
725     # recount
726     for row_idx,row in enumerate(simplex_tbl):
727         # first row, restrictions
728         if row_idx == 0:
729             continue
730         for item_idx,item in enumerate(row):
731             if row_idx == direction_row:
732                 if item_idx == direction_col:
733                     new_simplex_tbl[row_idx][item_idx] = 1
734                     divided = float(np.divide(
735                     simplex_tbl[row_idx][item_idx],
736                     simplex_tbl[direction_row][direction_col])

```

```

737         ))
738         if divided == float('nan') or math.fabs(divided) == 2
739             float('inf'):
740             new_simplex_tbl[row_idx][item_idx] = 0
741         else:
742             new_simplex_tbl[row_idx][item_idx] = round(divided,2
743                                                         ROUND_ACCURACY2
744                                                         )
745         continue
746     if item_idx == direction_col:
747         new_simplex_tbl[row_idx][item_idx] = 0
748         new_simplex_tbl[len(new_simplex_tbl)-1][item_idx] = 0
749         continue
750
751     divided = float(np.divide(
752         simplex_tbl[direction_row][item_idx],
753         simplex_tbl[direction_row][direction_col]
754     ))
755
756     if divided == float('nan') or math.fabs(divided) == float(2
757         'inf'):
758         new_simplex_tbl[row_idx][item_idx] = round(item,2
759                                                         ROUND_ACCURACY)
760     else:
761         new_simplex_tbl[row_idx][item_idx] = round(
762             item - (simplex_tbl[row_idx][direction_col]2
763                     *divided),
764             ROUND_ACCURACY
765         )
766
767     basis_left_col[direction_row-1] = cost_flat[direction_col]
768     basis_idx[direction_row-1] = direction_col
769
770     target_func = 0
771     for idx in range(len(simplex_tbl)-1):
772         if idx != 0:
773             target_func += new_simplex_tbl[idx][0]*basis_left_col[idx-2
774                                     1]
775     new_simplex_tbl[len(new_simplex_tbl)-1][0] = target_func
776
777     simplex_tbl = new_simplex_tbl
778
779     iteration += 1
780
781 if not SUCCESS:
782     if iteration == MAX_ITERATIONS:
783         report_write('Не найдено решения за ' +\
784                     str(MAX_ITERATIONS) +\
785                     ' шагов. Или что-то пошло не так 2
786                     или надо увеличить число шагов.')
787     else:
788         report_write('В направляющем столбце 2
789                     нет положительных 2
790                     элементов. Это значит, что 2
791                     целевая функция убывает.')
792
793 if SUCCESS:
794     print('done')
795
796 report_output.close()

```