

УДК 372.881.1

АВТОМАТИЗАЦИЯ ПРОЦЕССА СОСТАВЛЕНИЯ КОМПЛЕКСОВ УПРАЖНЕНИЙ ПО ГРАММАТИКЕ НЕМЕЦКОГО ЯЗЫКА НА ОСНОВЕ КРУПНОГО ХУДОЖЕСТВЕННОГО ПРОИЗВЕДЕНИЯ

Горожанов Алексей Иванович

канд. филол. наук
Московский государственный лингвистический университет, Москва

author@apriori-journal.ru

Аннотация. Рассматривается процесс составления комплекса упражнений по грамматике немецкого языка, используя в качестве языкового материала отрывки романа Э.М. Ремарка «Возлюби ближнего своего». С помощью авторского программного обеспечения текст романа преобразуется в корпус текстов, и с его помощью составляются упражнения пяти типов. Комплекс упражнений снабжается списком вокабуляра и карточками. В статье частично приводится программный код на Java, Python и HTML.

Ключевые слова: комплекс упражнений; грамматика немецкого языка; автоматизация; художественная литература; корпус текстов; Python; Java.

AUTOMATION OF COMPILING SETS OF EXERCISES IN GERMAN GRAMMAR BASED ON A LONG FICTIONAL NARRATIVE

Gorozhanov Alexey Ivanovich

candidate of philology
Moscow State Linguistic University, Moscow

Abstract. The paper touches upon automation of compiling sets of exercises in German Grammar based on the novel by E.M. Remarque «Liebe deinen Nächsten». Applying author's software the text of the novel is being transformed to a text corpus, and then a set of exercises is being compiled represented by five types of activities. The set of exercises is supplied with a list of vocabulary and flashcards. The programming code on Java, Python and HTML is partly included in the paper.

Key words: set of exercises; German language grammar; automation; fiction; text corpus; Python; Java.

Разработка комплексов упражнений по грамматике немецкого языка (как, впрочем, и по грамматике любого языка) является чрезвычайно трудоемким процессом, не в последнюю очередь в связи с необходимостью анализа больших объемов аутентичного языкового материала.

Современное программное обеспечение позволяет автоматизировать процесс отбора релевантных образцов (здесь и далее мы будем понимать под образцом некоторый объем аутентичного текста, достаточный для иллюстрации того или иного грамматического явления), позволяя автору сосредоточиться на решении методических задач.

Ниже мы опишем этапы автоматизированного процесса составления комплекса упражнений по грамматике немецкого языка при помощи автор-

ского программного обеспечения. Для создания тематической целостности в качестве языкового материала был выбран роман Э.М. Ремарка «Возлюби ближнего своего» (нем. «Liebe deinen Nächsten»).

На первом (подготовительном) этапе работы электронная версия романа была разбита на предложения при помощи программы для ЭВМ «Помощник лингвиста», позволяющей преобразовать текст в нумерованный список предложений, где критерием разделения являются сочетания символов «.!?» (точка, восклицательный знак, вопросительный знак) и пробела или символа новой строки. Программа «Помощник лингвиста» написана на языке программирования Java [1]. Функция, отвечающая за преобразование, выглядит следующим образом (см. Листинг 1):

Листинг 1

Код функции printSentence()

```
public static void printSentences(String text) {
    text = text.trim();
    char c = '\u0020'; // unicode w/space character
    for (int i=0; i<text.length(); i++) {
        if (Character.isWhitespace(text.charAt(i))) {
            text = text.replace(text.charAt(i), c);
        }
    }
    text = text.replace(".", "&&");
    text = text.replace("!", "!&&");
    text = text.replace("?", "?&&");
    // trim w/spaces between the words.
    while (text.contains(" ")) {
        text = text.replace(" ", " ");
    }
    String strRes = "Sentence layout:\n";
    // returns symbols . ! ? to normal
    String delim = "&&" + Character.toString(c);
    String[] str = text.split(delim);
    TextMain.lb_status.setText("The text contains " + str.length + "
sentence(s)");
    for (int i=0; i<str.length; i++) {
```

```

        str[i] = str[i].trim();
        str[i] = i+1 + ". " + str[i].replace("&&", "");
        strRes += str[i] + "\n";
        //System.out.println(str[i]);
    }
    TextMain.textArea_1.setText(strRes);
}

```

Тем самым весь текст разбивается на предложения достаточно точно, исключая ситуации, в которых после точки, восклицательного знака или вопросительного знака стоит какой-либо символ, не являющийся пробелом или символом новой строки, например, символ кавычек. В этом случае образец может включить в себя несколько предложений, например: «*Ist was los?*» *flüsterte er*. Однако, такую погрешность можно считать допустимой, т. к. впоследствии она будет скомпенсирована. В результате было получено 10420 образцов (при затрате времени около трех секунд!).

Полученные образцы были сохранены в текстовом файле с кодировкой «utf-8», при этом все кавычки были заменены на астериск (*). Эта работа была проведена в свободно распространяемой программе Notepad++, имеющей удобный инструмент поиска и замены текста [2]. В начало файла было помещено заглавие книги:

*Erich Maria Remarque *Liebe deinen Nächsten**
1. Kern fuhr mit einem Ruck aus schwarzem, brodelndem Schlaf empor und lauschte.
 ...
*10420. *Furchtbar viele Menschen.**

Затем этот текстовый файл был программно преобразован в специальный файл (базу данных) формата XML, который необходим для работы с образцами при помощи комплекта программ для ЭВМ «Помощник корпусного лингвиста». Описание структуры этого файла приведено ниже:

Первая строка выглядит так: `<?xml version="1.0" encoding="utf-8"?><book ttl="Erich Maria Remarque *Liebe deinen Nächsten*">`. Здесь указывается кодировка (utf-8) и заглавие текста (Erich Maria Remarque *Liebe deinen Nächsten*). Здесь же открывается тег «book», в котором будут находиться все остальные данные.

Далее, внутри тега «book», разместятся теги «s». В каждом из этих тегов будет находиться один образец, а в атрибутах тега – грамматические характеристики, которые пока что имеют значения по умолчанию, например: `<s aB="1" aC="1" aN="0" aP="1" aS="0" aU="1" gA="1" gP="0" gZ="0" id="1" m="1" mV="0" pA="0" pD="0" pDA="1" pG="0" pfT="1" pfU="0" prd="0" sV="0" st="2" tF1="0" tF2="0" tPf="0" tPs="1" tPt="0" tQ="0">Kern fuhr mit einem Ruck aus schwarzem, brodelndem Schlaf empor und lauschte.</s>`

Атрибуты располагаются внутри тега «s» в алфавитном порядке и расшифровываются следующим образом:

st – предложение простое (0) или сложное (сложносочиненное (1) и сложноподчиненное (2));

t – временная форма, дополнительные коды: tPs – презенс, tPf – перфект, tPt – претеритум, tQ – плюсквамперфект, tF1 – футур I, tF2 – футур II (принимают значения 0 или 1);

m – наклонение (индикатив (1), императив (2), конъюнктив (3));

mV – модальный глагол (0 или 1);

aN, aB, aU – 0 или 1 для нулевого, определенного или неопределенного артикля соответственно;

pG, pD, pA, pDA – предлог, управляющий генитивом, дативом, аккумулятивом или двойного управления (0 или 1);

pfT, pfU – отделяемая или неотделяемая приставка (0 или 1);

aP, aC, aS – прилагательное в позитиве, компаративе, суперлативе (0 или 1);

pred – предикатив (0 или 1);

gA, gP, gZ – залог (активный, пассивный, пассив состояния соответственно) (0 или 1);

sV – возвратный глагол (0 или 1) [3, с. 2-3].

Программа, в которой было выполнено преобразование текстового файла в файл XML, была написана на языке программирования Python и представляет собой класс со следующим кодом (см. Листинг 2):

Листинг 2

Код класса ReduceSent()

```
class ReduceSent():
    attrBlank = "st='0' tPs='0' tPf='0' tPt='0' tQ='0' tF1='0' tF2='0' m='1'
mV='0' aN='0' aB='0' aU='0' pG='0' pD='0' pA='0' pDA='0' pfT='0' pfU='0'
prd='0' aP='0' aC='0' aS='0' gA='1' gP='0' gZ='0' sV='0'"
    def __init__(self):
        self.file = open('Liebe deinen Naechsten.txt', 'r')
        self.lines1 = self.file.readlines()
        self.file.close()
        self.lines = []
        self.lines.append(self.lines1[0])
        for i in range(1, len(self.lines1)):
            if ('*' not in self.lines1[i]) and (len(self.lines1[i]) > 33):
                self.lines.append(self.lines1[i])
        self.file = open('111.xml', 'w')
        self.file.write("<?xml version='1.0' encoding='utf-8'?>")
        self.file.write("<book ttl='" + self.lines[0][0 : -1] + "'>\n")
        del self.lines[0]
        for i in range(len(self.lines)):
            #self.lineWN = self.line.split()
            #self.num = " " + self.lineWN[0][0 : -1]
            self.num = str(i + 1)
            self.lines[i] = "<s id='" + self.num + "' " + self.attrBlank + ">" +
self.lines[i][len(self.num)+2 : -1] + "</s>\n"
            self.file.write(self.lines[i])
        self.file.write("</book>")
        self.file.close()

if __name__ == '__main__':
    a = ReduceSent()
```

Приведенная в Листинге 2 программа не только приводит образцы к машиночитаемому виду, но и сокращает их количество по заданным критериям. Поскольку нашей целью является отбор языкового материала для иллюстрации грамматических явлений внутри контекста, мы отнесем к нерелевантным образцы, представляющие собой короткие эллиптические предложения, как например, реплики диалогов: 43. **Verflucht!* 44. *'raus!* 45. *Polski, 'raus!* Поэтому программа не включила в файл XML предложения, содержащие астериск (*) и образцы длиной менее 34 символов, включая пробелы. Таким образом, при средней длине образца равной 68 символам, минимальный порог в 34 символа составил около 50 % от длины среднего предложения рассматриваемого романа. Содержание файла XML было сокращено до 4569 образцов (то есть до 43,85 % от первоначального объема). На этом первый (подготовительный) этап был завершен.

Второй (основной) этап предполагал два варианта продолжения работы – исходя из функциональных возможностей комплекта программ для ЭВМ «Помощник корпусного лингвиста». Программа-менеджер, входящая в указанный комплект, способна работать как с размеченным, так и с неразмеченным корпусом текстов (представленным в виде файла XML). Полученный в ходе первого этапа работы корпус являлся неразмеченным (хотя и подготовленным к разметке). Размеченный корпус позволяет находить образцы, содержащие (или не содержащие, как «нулевой маркер») те или иные грамматические явления. С его помощью можно отбирать образцы, в которых, например, одновременно присутствуют конъюнктив и модальные глаголы или плюсквамперфект и глаголы с отделяемыми приставками. Однако, для этого необходимо произвести разметку с помощью специальной программы комплекта, рассматривая каждый образец в отдельности и включая в графическом интерфейсе программы соответствующие флажки и кнопки радио (см. Рис. 1):

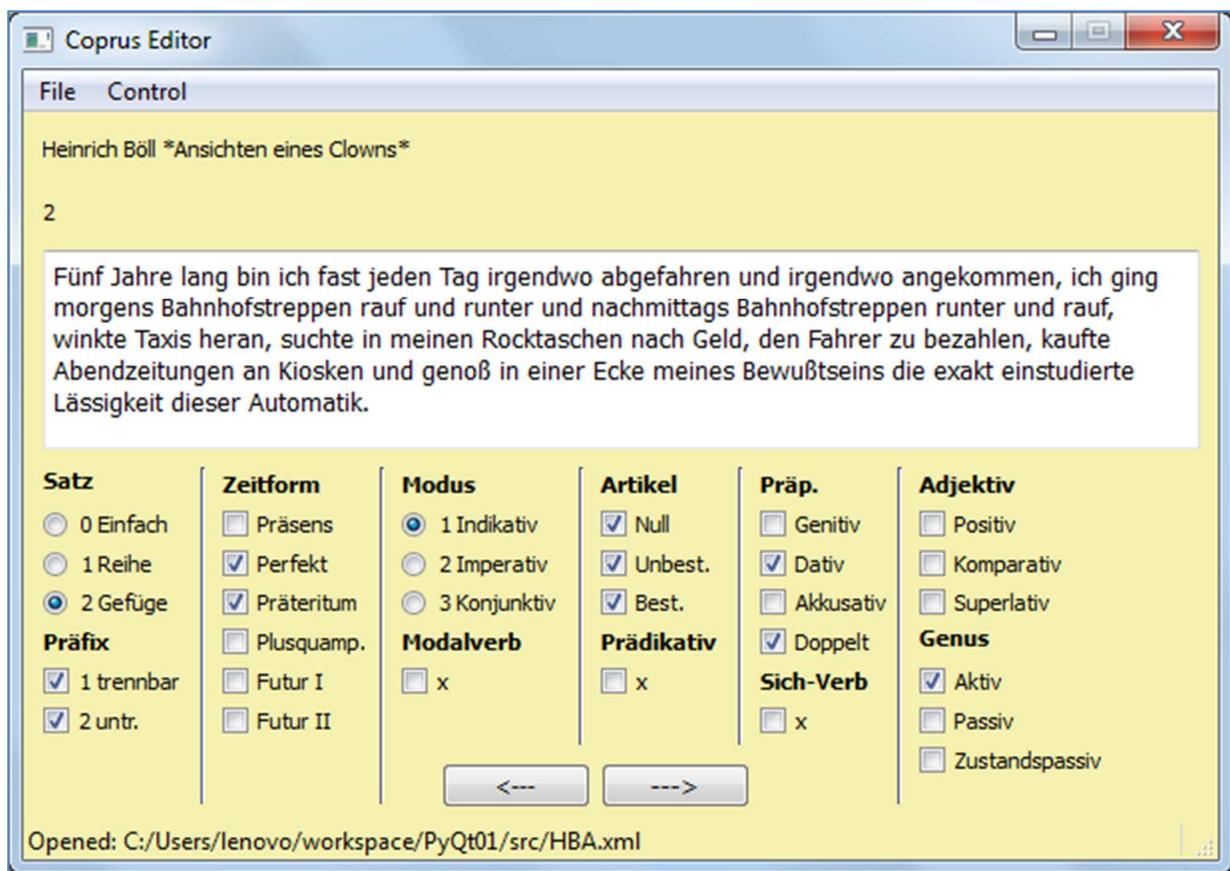


Рис. 1. Интерфейс программы разметки

В свою очередь, программа-менеджер допускает работу с неразмеченным корпусом, анализируя словоформы при помощи регулярных выражений PyQt5 [4]. Например, при вводе в поле поиска регулярного выражения $(\backslash b+(w\ddot{a}r)(\backslash w)\{1,3\}\backslash b+)|(\backslash b+(h\ddot{a}t\ddot{t}e)(\backslash w)\{1,3\}\backslash b+)$, программа выделит цветом фона следующие словоформы: *wäre, wärest, wäret, wären, hätte, hättest, hättest, hätten*. Нажатием кнопки «Reduce» из поля вывода будут удалены все образцы, не имеющие выделенных цветом элементов, то есть не содержащие указанные словоформы.

Безусловно, максимальный эффект достигается при использовании обеих возможностей поиска (с помощью грамматической разметки и регулярных выражений), все же в данной работе мы сфокусируемся на варианте работы с неразмеченным корпусом, как на менее трудоемком, но тем не менее достаточно эффективном.

Выбрав способ работы, мы приступили к процессу построения упражнений. В качестве примера приведем ниже разработку тему «Трудности употребления глагола *lassen*». Для получения эмпирического материала было подготовлено регулярное выражение

$(\backslash\text{b}+(\text{ge})?(l[\text{a}\ddot{\text{a}}]\text{ss})(\text{e}|\text{en}|\text{t})?\backslash\text{b}+)(\backslash\text{b}+(\text{lie}\ddot{\text{B}})(\text{e}|\text{en}|\text{t}|\text{est}|\text{et})?\backslash\text{b}+)$,

с помощью которого планировалось извлечь из корпуса образцы, содержащие словоформы *gelassen*, *lassen*, *lass*, *lasse*, *lasst*, *lässt*, *ließ*, *ließe*, *ließen*, *ließt*, *ließest*, *ließet*. Поиск по заданному регулярному выражению выявил 69 образцов, содержащих словоформы *lassen*, *ließ*, *gelassen* (причем вариант *gelassen* в образце 2472 оказался наречием, имеющим значение «хладнокровно»). Полученный материал послужил основой для составления комплекса упражнений пяти типов. Упражнения первого типа представляют собой определение значения глагола *lassen* и составлены при помощи соответствующей статьи из словаря Duden Online [5], например:

Aufgabe 1.1. In welchen Belegen fungiert «lassen» als Vollverb?

1. Es war am besten, den Mann in Ruhe zu lassen.
2. Er ließ seinen Koffer am Bahnhof und ging sofort zur Polizei.
3. Vielleicht hatte er sich den Bart inzwischen abnehmen lassen.
4. Er hatte seinen Rucksack dort in der Gepäckaufbewahrung gelassen.
5. Er glaubte, dass sie sich hatte scheiden lassen.
6. Kern ging am nächsten Vormittag mit Klassmann zur Flüchtlingshilfe, um sich einschreiben zu lassen.

Правильные ответы: 1, 2, 4.

Aufgabe 1.2. In welchen Belegen weist das Verb «lassen» Bedeutung «veranlassen, bewirken (dass etwas geschieht)» auf?

1. Er glaubte, dass sie sich hatte scheiden lassen.

2. Steiner zog eine Fotografie von sich aus der Tasche, die er in einem Automaten für einen Schilling hatte machen lassen.
3. Er wollte ein paar Adressen abklappern und Binding weiterschlafen lassen.
4. Vielleicht hatte er sich den Bart inzwischen abnehmen lassen.
5. Wenn du herauskommst, sollst du dich nach der Tschechei ausweisen lassen und zurückkommen.
6. Er ließ sich einen Stadtplan geben und zeichnete sich den Weg zu Ruths Wohnung auf.
7. Er ließ seinen Koffer an der Gepäckaufbewahrungsstelle.

Правильные ответы: 1, 2, 4, 5, 6.

Aufgabe 1.3. In welchen Belegen weist das Verb «lassen» Bedeutung «zulassen, erlauben; dulden; nicht an etwas hindern» auf?

1. Er glaubte, dass sie sich hatte scheiden lassen.
2. Er kam herein und ließ die Tür hinter sich offenstehen.
3. Dann ließ er sich erschöpft zurückfallen und schloss tief atmend die Augen.
4. Goldbach ließ die Hände sinken.
5. Kern ging am nächsten Vormittag mit Klassmann zur Flüchtlingshilfe, um sich einschreiben zu lassen.
6. Die Leute, die ihnen begegneten, blieben stehen und ließen sie schweigend vorübergehen.

Правильные ответы: 2, 3, 4, 6.

Второй тип включает упражнения на нахождение синонимов, например:

Aufgabe 2.1. Sagen Sie es anders. Benutzen Sie statt «lassen» ein Synonym:

1. Es war am besten, den Mann in Ruhe zu lassen (z. B. Es war am besten, den Mann weiter nicht zu stören).

2. Er ließ sich einen Stadtplan geben und zeichnete sich den Weg zu Ruths Wohnung auf (z. B. Er bat um einen Stadtplan und zeichnete sich den Weg zu Ruths Wohnung auf).
3. Sie können mich noch einmal verhören lassen (z. B. Sie können befehlen, mich noch einmal zu verhören).
4. Steiner ließ das Fenster herunter und sah hinaus (Steiner öffnete das Fenster und sah hinaus).

Третий тип упражнений предполагает выполнение грамматических трансформаций, например:

Aufgabe 3.1. Verwenden Sie die folgenden Aussagen im Perfekt:

1. Goldbach ließ die Hände sinken (Goldbach hat die Hände sinken lassen).
2. Er ließ sich einen Stadtplan geben und zeichnete sich den Weg zu Ruths Wohnung auf (Er hat sich einen Stadtplan geben lassen und sich den Weg zu Ruths Wohnung aufgezeichnet).
3. Steiner zog eine Fotografie von sich aus der Tasche, die er in einem Automaten für einen Schilling hatte machen lassen (Steiner hat eine Fotografie von sich aus der Tasche gezogen, die er in einem Automaten für einen Schilling hat machen lassen).
4. Er ließ seinen Koffer am Bahnhof und ging sofort zur Polizei (Er hat seinen Koffer am Bahnhof gelassen und ist sofort zur Polizei gegangen).

Четвертый тип упражнений предусматривает открытые задания, например, написание комментария:

Aufgabe 4.1. Kommentieren Sie den Gebrauch von «lassen» in den folgenden Aussagen:

1. Kern ging am nächsten Vormittag mit Klassmann zur Flüchtlingshilfe, um sich einschreiben zu lassen.

Возможный комментарий: *Das Verb «lassen» fungiert in dieser Aussage als Modalverb mit der Bedeutung «veranlassen, bewirken (dass etwas geschieht)». Es wird mit dem Vollverb «sich einschreiben» gebraucht. Das Verb «lassen» steht im Infinitiv mit der Konjunktion «zu» als Teil der Infinitivgruppe (mit «um» eingeleitet).*

Пятый тип упражнений предусматривает перевод с немецкого на русский язык, например:

Aufgabe 5.1. Übersetzen Sie die folgenden Aussagen. Achten Sie dabei auf die Bedeutung von «lassen»:

1. Es war am besten, den Mann in Ruhe zu lassen (например, «Лучше всего было оставить человека в покое»).
2. Er ließ seinen Koffer am Bahnhof und ging sofort zur Polizei (например, «Он оставил свой чемодан на вокзале и немедленно направился в полицию»).
3. Vielleicht hatte er sich den Bart inzwischen abnehmen lassen (например, «Может быть, он за это время сбрил бороду»).
4. Er hatte seinen Rucksack dort in der Gepäckaufbewahrung gelassen (например, «Там он оставил свой рюкзак в камере хранения»).
5. Er glaubte, dass sie sich hatte scheiden lassen (например, «Он полагал, что она уже подала на развод»).
6. Kern ging am nächsten Vormittag mit Klassmann zur Flüchtlingshilfe, um sich einschreiben zu lassen (например, «На следующее утро Керн отправился с Классманом в бюро помощи эмигрантам, чтобы зарегистрироваться»).

Поскольку использованные в упражнениях образцы тематически связаны (являются отрывками одного и того же романа), грамматические упражнения были дополнены списком вокабуляра, чтобы процесс

изучения грамматики не превратился только в заучивание отдельных словоформ. Процесс составления вокабуляра к упражнениям также был автоматизирован.

С помощью программы «Помощник лингвиста» был составлен список частотности словоформ образцов, содержащих глагол *lassen*. В список, отсортированный по убыванию параметра частотности, вошли 133 словоформы. Поскольку самыми частотными при таком анализе являются союзы, артикли, местоимения и предлоги, например, *und* – 73, *er* – 50, *die* – 32, *sich* – 31 и т. д., эти словоформы были исключены из списка. Также из него были исключены глагол *lassen*, модальные глаголы и имена собственные. В результате список был сокращен до следующих 47 единиц (в алфавитном порядке, после двоеточия приведено количество употреблений):

1. Arm : 2
2. auch : 2
3. Augen : 2
4. Beamte : 2
5. beide : 2
6. Bett : 3
7. Bild : 3
8. Boden : 4
9. etwas : 2
10. fallen : 2
11. fassen : 2
12. Fenster : 3
13. Fotografie : 2
14. Gesicht : 5
15. Grenze : 2
16. Hand : 5
17. holen : 2
18. immer : 2
19. Jahr : 4
20. Koffer : 3
21. kommen : 2

- 22. langsam : 4
- 23. Leben : 3
- 24. legen : 2
- 25. Licht : 2
- 26. Liebe : 2
- 27. machen : 2
- 28. Mann : 2
- 29. Mantel : 3
- 30. Nacken : 2
- 31. plötzlich : 3
- 32. Schatte : 2
- 33. scheiden : 2
- 34. Schraubenzieher : 2
- 35. schweigen : 2
- 36. schwer : 2
- 37. sehen : 2
- 38. selbst : 2
- 39. sinken : 2
- 40. sofort : 2
- 41. stark : 2
- 42. sterben : 2
- 43. tief : 2
- 44. Wand : 2
- 45. Wärme : 4
- 46. wissen : 2
- 47. Zigarette : 2

Как видно из списка, вокабуляр является базовым и соответствует, в целом, уровню А1. Поэтому в дополнение к включению в комплекс упражнений приведенного списка (в виде справочной информации), было решено использовать в качестве инструмента обогащения лексического запаса обучающихся отобранные образцы целиком.

С этой целью 68 образцов были программно сохранены в файл HTML таким образом, чтобы после вывода файла на печать его бумажную копию можно было бы разрезать и получить карточки, на лицевой стороне каждой из которых находилось бы по одному образцу.

Для получения такого файла отобранные программой-менеджером образцы были сохранены в формате HTML (для этого предусмотрен специальный пункт меню). Далее файл был преобразован для того, чтобы получить вид карточек. В исходном виде он содержал следующий код (см. Листинг 3):

Листинг 3

Код исходного файла HTML до преобразования (сокращенно)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
  <html><head><meta name="qrichtext" content="1" /><style
type="text/css">
  p, li { white-space: pre-wrap; }
  </style></head><body style=" font-family:'MS Shell Dlg 2'; font-size:10pt;
font-weight:400; font-style:normal;">
  <p style=" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;">20. Der Mann im Hemd ließ
den Lichtkreis hinüberwandem.</p>
  <p style="-qt-paragraph-type:empty; margin-top:0px; margin-bottom:0px;
margin-left:0px; margin-right:0px; -qt-block-indent:0; text-indent:0px;"><br
/></p>
  <footer align='right'><em>E.M.Remarque. Liebe deinen
Nächsten</em></footer></body></html>
```

Программа-преобразователь была написана на Python (см. Листинг 4):

Листинг 4

Код программы-преобразователя

```
import re
f = open('subjects.html', 'r', encoding='utf8')
text = f.read()
f.close()
text = text.replace('<style type="text/css">', '<style type="text/css">\np {
font-size: 13pt; }\ntd { border: 1pt dashed; height: 170px; width: 300px; }')
text = re.sub('<footer(.*?)footer>', '', text)
```

```

text = re.sub('style="(.*)"', "", text)
text = re.sub('<br(.*)/>', "", text)
text = re.sub('<p ></p>', "", text)
text = text.replace('<body >', '<body>\n<table>')
text = text.replace('</body>', '</table>\n</body>')
text = re.sub('<p >', '<tr><td align="center" valign="center">', text)
text = re.sub('</p>', '</td><td></td></tr>', text)
f = open('subjects1.html', 'w', encoding='utf8')
f.write(text)
f.close()

```

Каждый образец помещается внутри ячейки таблицы с горизонтальным и вертикальным выравниванием по центру. Высота и ширина ячейки являются фиксированной: 170 и 300 пикселей соответственно. Рядом с заполненной ячейкой располагается такая же (но незаполненная), которая будет являться ее обратной стороной. Разделители выполнены тонкой прерывистой линией для экономии ресурсов принтера.

В результате была получена таблица, каждый ряд которой представляет собой двустороннюю карточку (см. Рис. 2):

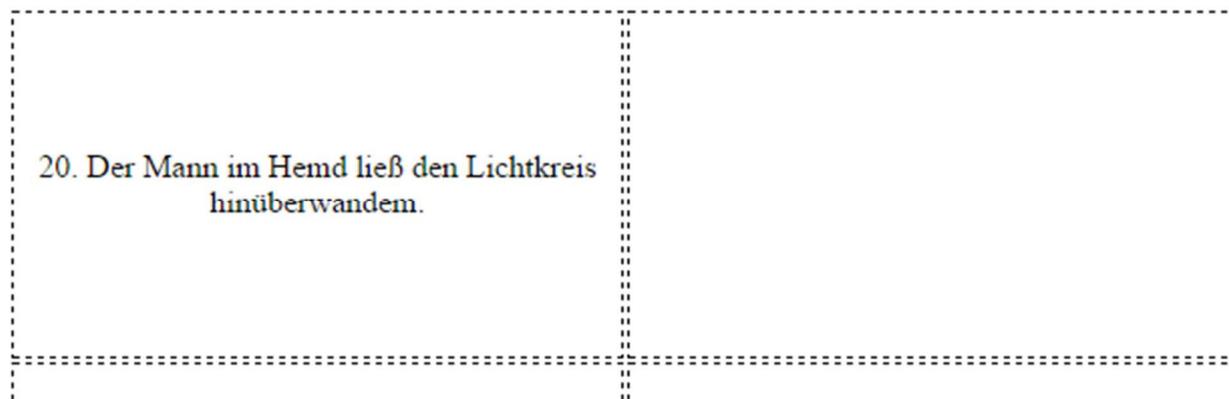


Рис. 2. Фрагмент таблицы карточек

С полученными карточками можно работать различными способами. Самый простой из них – распечатать файл HTML, разрезать распечатку на фрагменты (как на Рис. 2), согнуть по вертикальной линии, склеить и записать какую-либо информацию на чистой стороне (перевод, коммен-

тарий, грамматические характеристики и т. п.). Также можно скопировать содержимое файла HTML (выделить при помощи сочетания клавиш «Ctrl + а» и затем нажать «Ctrl + с»), поместить его в текстовый процессор (например, MS Word) и заполнить чистую сторону карточек каким-либо печатным текстом, далее распечатать, разрезать и склеить.

В результате мы получили тематически связанный комплекс упражнений по грамматике немецкого языка (тема: «Трудности употребления глагола lassen»), включающий упражнения пяти типов, список вокабуляра и карточки. Процесс был максимально автоматизирован при помощи специального программного обеспечения, которое выполнило всю рутинную работу, позволив автору сосредоточиться на таких проблемах, как выбор грамматической темы, выбор типов упражнений и компоновка материала внутри упражнений. Ручная корректировка полученных в результате работы программ образцов была минимальной. При этом программно был обработан большой объем текста (роман Э.М. Ремарка «Возлюби ближнего своего» состоит из более чем 600 тыс. печатных знаков без учета пробелов) при затратах времени не более нескольких секунд на одну операцию.

Таким образом, представленный нами процесс составления комплексов упражнений в целом – и использование регулярных выражений в программе-менеджере комплекта «Помощник корпусного лингвиста» для работы с неразмеченными корпусами текстов в частности – показали себя как высокоэффективные, но в то же время только *относительно* универсальные методы, т.к. их внедрение предполагает наличие у преподавателя-автора элементарных навыков профессионально ориентированного программирования (примерный набор задач обработки текста, решаемых на языках программирования Java и Python приведен нами в «Задачнике по сопоставительному программированию: Java vs. Python» [6]). По-настоящему универсальными эти методы могут стать только в том случае, если автор будет одновременно являться и специ-

алистом по методике преподавания иностранного языка, и профессионально ориентированным программистом.

Полученные комплексы упражнений, ввиду тематической связности языкового материала, могут быть использованы не только на занятиях по грамматике иностранного языка, но также при реализации таких дисциплин как «Домашнее чтение», «Перевод», «Анализ текста» и др.

Список использованных источников

1. Горожанов А.И. Linguist's Companion – an example of educational software // Слово. Предложение. Текст: анализ языковой культуры: Матер. V Междунар. науч.-практ. конф. (25 февраля 2014 г.). Краснодар, 2014. С. 181-187.
2. Notepad Plus Plus Home Page [Электронный ресурс]. Режим доступа: <http://notepad-plus-plus.org> (дата обращения: 22.04.2015).
3. Горожанов А.И. Комплект программ для ЭВМ «Помощник корпусного лингвиста» // Библиотека РТО на портале ОФЭРНиО. 07.10.2014. [Электронный ресурс]. Режим доступа: http://ofernio.ru/rto_files_ofernio/20413.doc (дата обращения: 22.04.2015).
4. QRegExp Class / Qt Documentation [Электронный ресурс]. Режим доступа: <http://doc.qt.io/qt-5/qregexp.html> (дата обращения: 23.04.2015).
5. Duden Online [Электронный ресурс]. Режим доступа: <http://www.duden.de/rechtschreibung/lassen> (дата обращения: 23.04.2015).
6. Горожанов А.И. Электронное учебное пособие «Задачник по сопоставительному программированию: Java vs. Python» // Библиотека РТО на портале ОФЭРНиО. 07.10.2014. [Электронный ресурс]. Режим доступа: http://ofernio.ru/rto_files_ofernio/20561.doc (дата обращения: 26.04.2015).