



УДК: 004.9

© 2022 г. А.А. Дедегкаева, канд. техн. наук,

**А.В. Горбунов**

(Северо-Кавказский горно-металлургический институт  
«государственный технологический университет», Владикавказ)

## **МЕТОД ЛЕЙТНЕРА В МОБИЛЬНОМ ПРИЛОЖЕНИИ**

Статья посвящена формализации метода Лейтнера для разработки мобильного приложения, предназначенного для заучивания иностранных слов с помощью систематических уведомлений, корректирующего параметры своей работы в зависимости от поведения пользователя.

**Ключевые слова:** метод Лейтнера, мобильное приложение, подмножество, нечеткое множество, функция принадлежности, показатель сложности слова, автоматный граф, переходы автомата.

DOI: 10.22250/18142400\_2022\_71\_1\_101

### **Введение**

Ввиду многообразия методов изучения английского языка и сложности формализации факторов, влияющих на успешное применение этих методов, задача их формализации остается актуальной, несмотря на большое количество программных приложений для изучения языка.

В данной статье предлагается формализация метода Лейтнера для разработки мобильного приложения, в котором функцию карточек выполняют уведомления пользователю, содержащие слово, транскрипцию и перевод, действующие в качестве раздражителя на мозг. Такое приложение позволяет пользователям заучивать английские слова без особых усилий и затрат времени.

### **Метод Лейтнера**

Забывание – неизбежное свойство человеческой памяти. Множество психологических исследований посвящено изучению памяти и процесса забывания. Результаты этих исследований позволяют определить наиболее

эффективные методы запоминания. Один из них – метод карточек, предложенный немецким ученым и журналистом Себастьяном Лейтнером в 70-е гг. прошлого века [1].

Метод заключается в повторении иностранных слов через разные промежутки времени, в зависимости от результата воспроизведения слова по памяти. Система Лейтнера предполагает три коробки под флеш-карточки. В первую ячейку кладутся карточки с новыми и сложно запоминающимися иностранными словами. Вторая ячейка должна содержать карточки со словами, которые были названы неправильно. Третья ячейка содержит карточки с легко воспроизводимыми по памяти словами.

Флеш-карточки из первой ячейки должны использоваться один раз в день. Во вторую ячейку надо поместить слова, которые не требуют частого повторения, – скажем, с перерывом в один день. В третью ячейку перекладываются хорошо заученные слова из первых двух ячеек. Если слова на карточках из третьей ячейки частично подверглись забыванию, их стоит переместить обратно во вторую. Если слова на флеш-карточках второй ячейки вспоминаются с ошибками, то их следует переместить в первую ячейку и повторять как новые.

### **Описание модели метода Лейтнера**

Карточке ставится в соответствие последовательность (слово, транскрипция, перевод). Будем называть ее словом и обозначать  $w$  ( $w \in W$ ,  $W$  – множество слов, подлежащих изучению).

Каждому слову ставится в соответствие значение от одного до пяти ( $K$ ), которое соответствует «номеру коробки» и определяет частоту отображения слова. Первоначально для всех слов  $K = 0$ . Далее выделяется подмножество слов, с которых начнется изучение, для них  $K$  устанавливается равным единице, «слова попадают в первую коробку». Множество слов с определенным значением  $K$ , «находящихся в одной коробке», будем обозначать  $W_K$ .

Перемещение «карточек» между «коробками» осуществляется по результату тестирования, который может быть представлен в виде двух подмножеств множества слов, подлежащих тестированию ( $W_T$ ): подмножество  $W_R \subseteq W_T$ , содержащее слова, на которые был дан правильный ответ, и подмножество  $W_W \subseteq W_T$ , содержащее слова, на которые был дан неправильный ответ.

Отличные от нуля значения  $K$  изменяются в результате тестирования. Если слово было переведено в тесте правильно, то соответствующее ему

значение увеличивается на единицу, в противном случае уменьшается на единицу; значение, равное пяти, остается без изменения. Если после очередного тестирования найдется слово, для которого  $K$  изменилось с четверки на пятерку, то для одного из слов, принадлежащих  $W_0$ , значение меняется на единицу.

Изменение  $K$  для определенного слова  $w$  можно проиллюстрировать с помощью графа, представленного на рис. 1.

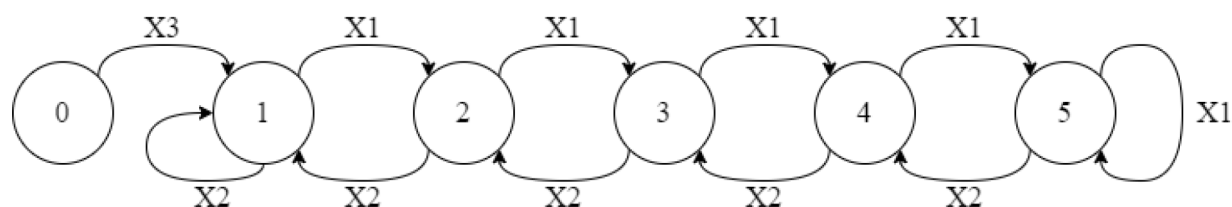


Рис. 1. Изменение значения  $K$ .

Вершины графа соответствуют значениям  $K$ , а веса дуг – результатам тестирования:

$X_1 - w \in W_R$ ;

$X_2 - w \in W_W$ ;

$X_3 - \exists w' (w' \in W_W \ \& \ K' = 5)$ .

### Определение количества демонстраций слова

Как уже было сказано, процесс тестирования инициируется пользователем, поэтому в каждый момент работы программы должна быть возможность определить множество слов, подлежащих тестированию. При этом следует учесть, что проверять слова, которые были просмотрены заведомо недостаточное количество раз, не следует. Это может привести к неоправданному уменьшению значения  $K$  и, как следствие, увеличению частоты отображений слова, что нарушает выбранную методику и будет только раздражать пользователя.

Поэтому предлагается использовать счетчик ( $N$ ), значение которого после каждого отображения слова увеличивается на единицу и обнуляется при изменении значения  $K$ . Слово принадлежит  $W_T$ , если соответствующее ему значение  $N$  достигло некоторого порогового значения. Для определения порогового значения воспользуемся следующими соображениями.

Во-первых, известно, что количество повторений, необходимое для заучивания слова, зависит в том числе от длины слова. Чем оно короче, тем легче запоминается. Поэтому каждому слову поставлено в соответствие значение  $L$ , характеризующее условную сложность слова следующим образом:

$$L = \begin{cases} 1, & \text{если } l \leq 4, \\ 2, & \text{если } 4 \leq l \leq 8, \\ 3, & \text{если } l \geq 8, \end{cases}$$

где значения 4 и 8 рекомендуются на основе опыта преподавателей английского языка и могут быть изменены по усмотрению разработчика приложения.

Такое соответствие разбивает множество  $W$  на три подмножества ( $W_L$ ). Пороговое значение будем определять для каждого из них отдельно.

Во-вторых, слова, у которых значение  $K$  больше, демонстрируются реже и их следует включать в тест после меньшего количества просмотров, т.е. пороговое значение зависит не только от сложности слова, но и от стадии его заучивания.

Таким образом, пороговое значение ( $N_{LK}$ ) требуется определить для каждого из подмножеств  $W_{LK} = W_K \cap W_L$ .

На основе работ [2] рекомендуется устанавливать первоначальные значения  $N_{LK}$  равными значениям, приведенным в таблице.

L \ K	1	2	3
1	2	3	4
2	3	4	5
3	4	5	6
4	5	6	7
5	6	7	8

Помимо длины слова, на количество повторений, необходимых для запоминания слова, влияет много других факторов: наличие созвучных слов в родном языке, латинские корни в словах, специальные термины, заимствованные слова и т.д. Для более точной классификации слов по сложности запоминания может быть применен опыт пользователей приложения.

С этой целью предлагается накапливать и анализировать данные о количестве повторений, потребовавшихся пользователям системы для запоминания слова. Так как это количество зависит не только от слова, но и от способностей конкретного пользователя, то собирать информацию предлагается в виде нечетких подмножеств «сложных слов теста»  $W_i = \{w, \mu_i(w) \mid w \in W_R \cap W_{K=1}\}$ , функция принадлежности которых определяется в виде

$$\mu_i(w) = N_w / N_{\max},$$

где  $N_w$  – количество повторений, выполненных для слова  $w$ ;  $N_{\max}$  – максимальное количество повторений для слов  $w \in W_R \cap W_{K=1}$ .

Тогда можно определить нечеткое множество «сложных слов»:

$$W_U = \{w, \mu_U(w) \mid w \in U\},$$

где  $U$  – множество слов изучаемого языка;  $\mu_U(w)$  первоначально принимается равным единице и корректируется после получения каждого  $W_i$  с помощью одной из операций пересечения нечетких множеств, – например, по формуле:

$$\forall w \in U \quad \mu_U(w) = (\mu'_U(w) + \mu_i(w)) / 2,$$

где –  $\mu'_U(w)$  значение  $\mu_U(w)$  до очередной корректировки.

Значения функции принадлежности полученного таким образом нечеткого множества могут быть использованы для классификации слов, т.е. в качестве значений  $L$  [3].

### **Корректировка значений $N_{LK}$ в соответствии с поведением пользователя**

Очевидно, что люди обладают разной способностью к запоминанию. Если значения  $N_{LK}$  окажутся слишком маленькими для конкретного пользователя, то  $K$  будет чаще уменьшаться, слова будут дольше «проходить путь от первой коробки до последней» и время, потраченное на заучивание, неоправданно возрастет. Если значения  $N_{LK}$  окажутся слишком большими, то слова будут дольше «задерживаться в одной коробке», это приведет к тем же последствиям. К тому же пользователя могут раздражать как отображение слов, которые он уже запомнил, так и тесты, в которых слишком много слов, еще не запомнившихся, и он попросту откажется от использования приложения.

Кроме того, известно, что по мере увеличения количества уже освоенных слов процесс запоминания новых слов ускоряется. Это связано с тем, что, зная базовые корни, легче запоминать слова с этими корнями. Например, слова «красивый», «прекрасный», «красотка» легче запомнить, если уже знаешь слово «красота». Через заучивание слов также усваиваются правила образования этих слов, поэтому чем больше слов пользователь уже освоил, тем меньше повторений ему требуется для запоминания новых слов.

Чтобы процесс заучивания слов с помощью приложения был комфортным и эффективным, значение  $N_{LK}$  следует менять в соответствии со способностями конкретного пользователя и стадией изучения языка.

Предлагаемый способ корректировки значений  $N_{LK}$  можно проиллюстрировать автоматным графом [4], представленным на рис. 2.

Каждая группа состояний соответствует некоторому значению  $N_{LK}$ . Входной сигнал  $X1$  соответствует правильному ответу пользователя, а  $X2$  – неправильному.

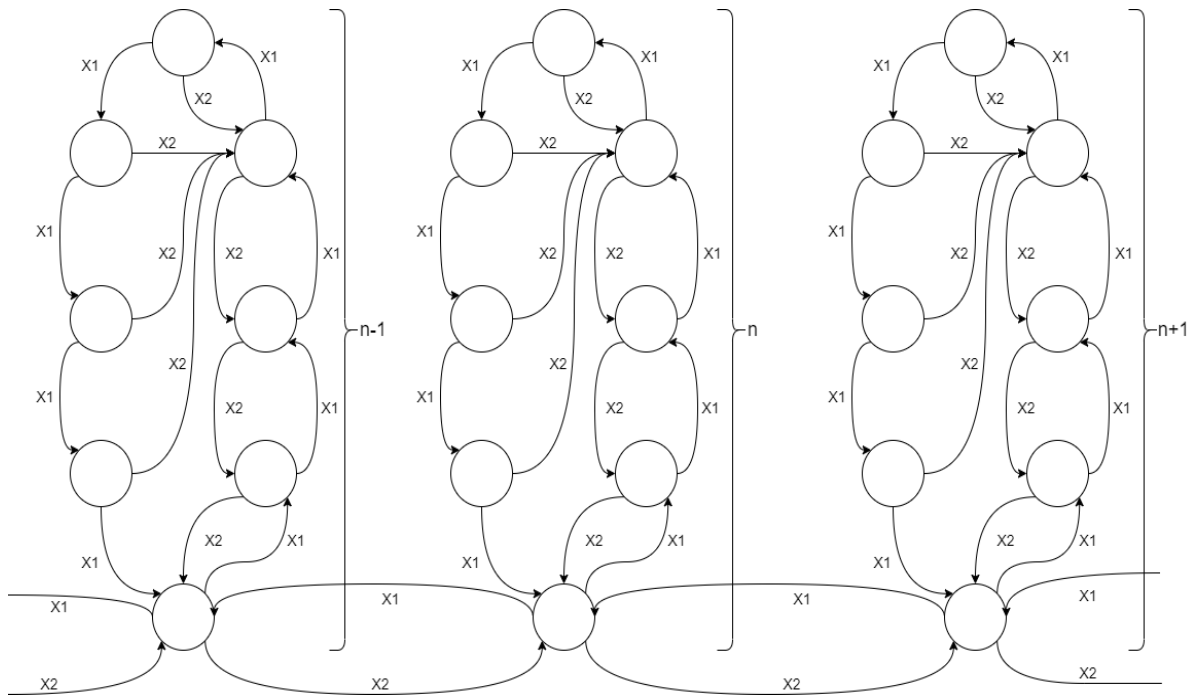


Рис. 2. Графовая модель способа корректировки значений  $N_{LK}$ .

После очередного такта работы автомата  $N_{LK}$  присваивается значение, соответствующее группе, содержащей состояние, в которое перешел автомат. Таким образом, если правильные и неправильные ответы чередуются, то значение  $N_{LK}$  условно соответствует потребностям пользователя и остается неизменным. В том случае, когда пользователь дает неправильные ответы чаще чем правильные, значение увеличивается. Если же пользователь дает подряд много правильных ответов, то можно предположить, что количество выполняемых повторений слов избыточно, значение уменьшается.

Количество состояний в группе определяет инертность значения  $N_{LK}$  и может быть задано параметрически.

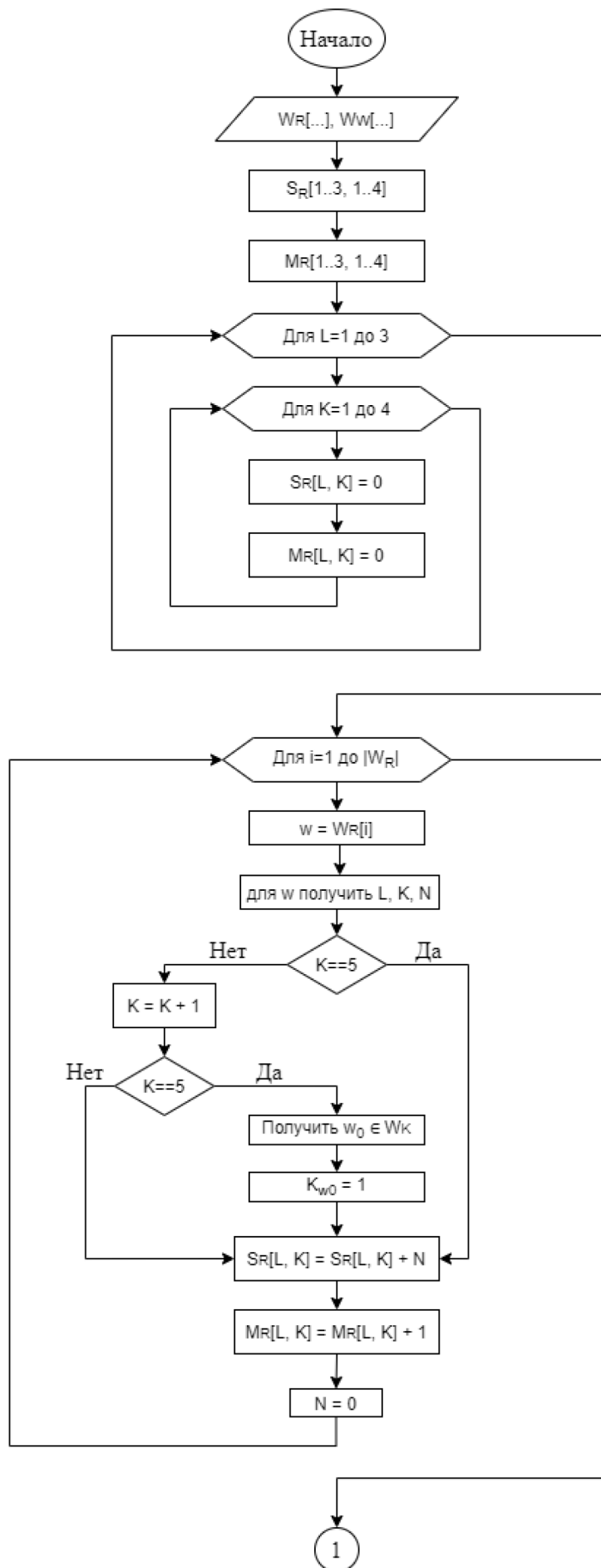
### Описание работы системы

Итак, в каждый момент работы системы:

каждому слову соответствует набор значений  $(L, K, N, t)$ , где значение  $N$  изменяется после демонстрации слова и при изменении  $K$ , причем значение  $K$  изменяется после обработки результатов тестирования;  $t$  – дата и время последней демонстрации слова, которые фиксируются после каждой демонстрации;

каждому значению  $K$  соответствует значение  $T_K$  – частота демонстрации слов;

каждой паре значений  $(L, K)$  соответствует значение  $(N_{LK})$ , которое изменяется после обработки результатов тестирования. Блок-схема алгоритма процедуры обработки тестирования представлена на рис. 3.



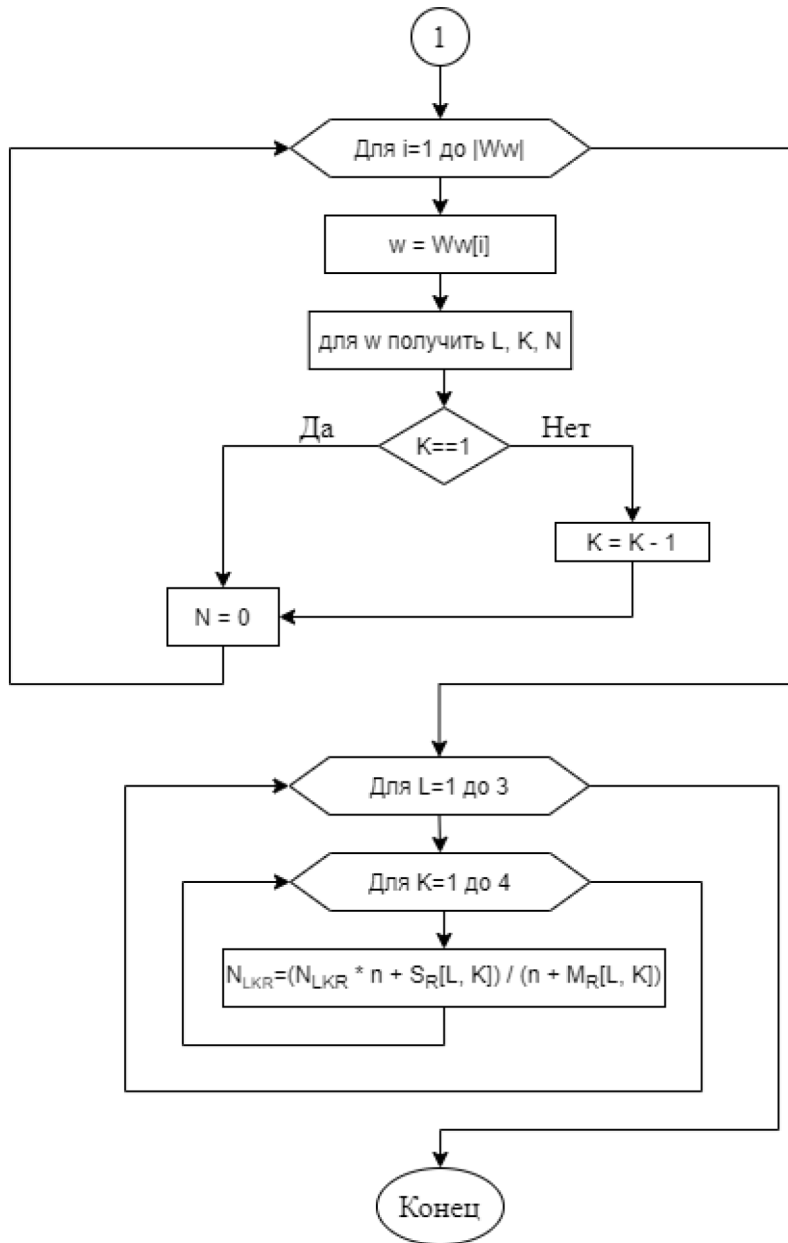


Рис. 3. Блок-схема алгоритма обработки тестирования.

Работа системы представлена на рис. 4.

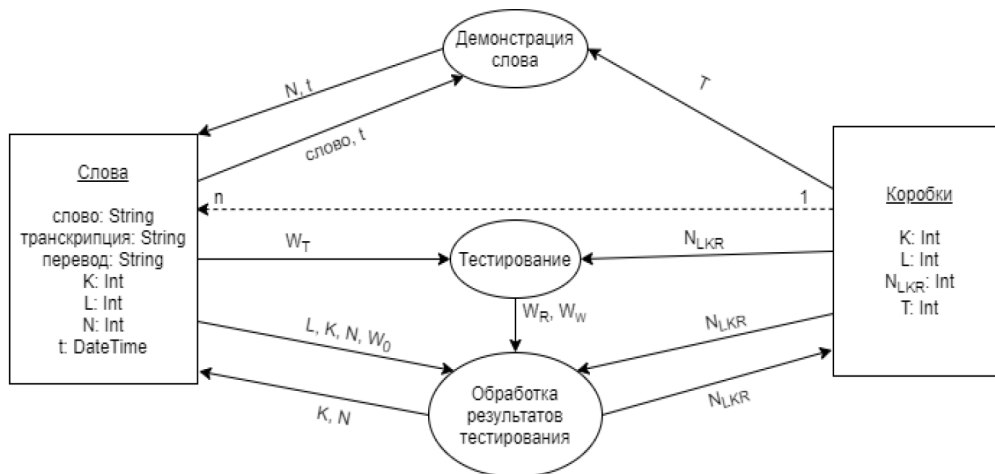


Рис. 4. Схема работы системы.



## Заключение

Проанализировав множество приложений для изучения английского языка, можно сделать вывод, что все они требуют от пользователя обязательного пребывания. Приложение, разработанное на основе, предложенной в данной статье формализации метода Лейтнера, позволит пользователю запоминать слова, не отводя для этого специального времени. Кроме того, изменив наполнение базы данных, такое приложение можно использовать не только для заучивания иностранных слов, но и любых других соответствий, – например, плана счетов, дорожных знаков и т.д.

## ЛИТЕРАТУРА

1. Метод использования карточек для длительного запоминания по системе Лейтнера (Leitner system) [Электронный ресурс]. – Режим доступа: <https://blog.wikium.ru/metod-ispolzovaniya-kartochek-dlya-dlitelnogo-zapominaniya-po-sisteme-leitnera-leitner-system.html>.
2. Блонский П.П. Память и мышление. – М.-Л.: СОЦЭКГИЗ, 1982.
3. Аверкин А.Н. Нечеткие множества в моделях управления и искусственного интеллекта. – М.: Книга по требованию, 2021.
4. Мелихов А.Н. Ориентированные графы и конечные автоматы. – М.: Наука, 1971.

*Статья представлена к публикации членом редколлегии Е.А. Шеленком.*

*E-mail:*

*Дедежкаева Анна Альбертовна – [dedegkaev\\_ag@mail.ru](mailto:dedegkaev_ag@mail.ru);*

*Горбунов Алан Владимирович – [alan.gorbunov@mail.ru](mailto:alan.gorbunov@mail.ru).*