

**НАВЧАЛЬНА ПРОГРАМА ІЗ ЕЛЕМЕНТАМИ ГРИ
«СЛОВНИЧОК»
НА MOBI BORLAND DELPHI 7**

ЗМІСТ

ВСТУП	3
РОЗДІЛ I СТРУКТУРНІ КОМПОНЕНТИ ПРОГРАМИ «СЛОВНИЧОК», СТВОРЕНОЇ НА МОВІ BORLAND DELPHI 7	5
1.1 Основні елементи навчальної програми «Словничок».....	5
1.2 Гра, як структурний компонент навчальної програми «Словничок».....	15
РОЗДІЛ II СПЕКТР ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ ПРОГРАМИ «СЛОВНИЧОК».....	22
2.1 Основні функціональні можливості створеної програми.....	22
2.2 Інтерфейс гри навчальної програми «Словничок»	30
ВИСНОВКИ.....	30
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	32

ВСТУП

Інформаційні технології – вимога сьогодення, що дозволяє створити суспільство, засноване на знаннях. Новітні інформаційні технології стрімко ввірвалися в усі сфери нашого життя, стали такою ж реальністю, як телефонний зв'язок чи подорожування літаком. Вони спрощують спілкування та співробітництво. Суспільство, яке дбає про своє майбутнє, має усвідомити колосальні можливості, привнесені новими інформаційними технологіями, та навчитися грамотно застосовувати їх, у першу чергу, в освіті.

Знання англійської мови є такою ж актуальною вимогою сьогодення. [5]

Існує багато програм, за допомогою яких людина може вивчати англійську мову. Всі вони мають, звичайно, власну методику та програму навчання. Але навчальна програма вивчення іноземної мови, на нашу думку, має свої недоліки: вивчення слів відбувається лише відповідно до неї. Спочатку вам пропонується вивчити певні слова. Потім програма перевіряє ваші знання, а далі пропонується побудувати декілька речень з використанням цих слів і т.д.

Але що робити, коли більшість слів, запропонованих навчальною програмою або знайомі, або зовсім не потрібні в даний момент? Знайомі слова повторюються, а непотрібні, зважаючи на те, що немає де їх застосувати на практиці, просто забуваються.

Тому виникає необхідність у створенні програми, яка б допомагала вивчати слова, які вам потрібні саме в цей момент.

Все це зумовило необхідність створення програми, доступної і цікавої для учнів при їх самостійному вивченні англійської мови, розробити наочне програмне забезпечення, яке буде викликати зацікавлення учнів, для кращого

запам'ятовування нового матеріалу. Результатом нашої роботи стала «Навчальна програма із елементами гри «СЛОВНИЧОК» на мові Borland Delphi 7», яка дозволяє:

- 1) зберігати слово, транскрипцію та переклад у власну базу даних;
- 2) полегшувати та прискорювати процес вивчення незнайомих слів;
- 3) спрощувати пошук слів та групувати їх відповідно до запитів користувача.

Методика вивчення англійської мови в програмі досить проста: ви вводите слова, знання яких Вам потрібно застосувати на практиці, а програма допомагає їх швидше вивчити.

Методика даної програми розрахована на людей, які вже мають певний багаж знань. Користувач повинен володіти базовими знаннями граматики англійської мови. Спеціалізація даної програми – вивчення слів.

Наведемо приклад: учень, вирішивши більш глибоко вивчати іноземну мову, придбав книжку англійською мовою. Він читає перший розділ і виписує незнайомі йому слова. Далі записує їх у «Словничок» і починає їх вивчати. Потім учень починає читати наступний розділ. Звичайно ж, він неодмінно зустрінеться зі словами, що були незнайомі для нього в першому розділі. Таким чином, з'являється практика використання слів. Прочитавши певну кількість розділів, учень має змогу повторити всі нові слова з даної книжки, цим самим затвердити їх знання.

РОЗДІЛ І СТРУКТУРНІ КОМПОНЕНТИ ПРОГРАМИ «СЛОВНИЧОК», СТВОРЕНОЇ НА MOBI BORLAND DELPHI 7

1.1 Основні елементи навчальної програми «Словничок»

Навчальна програма «Словничок» з елементами гри створена як альтернативний варіант ведення словника при вивченні іноземної мови.

Методика вивчення англійської мови в нашій програмі проста: користувач вводить слова, знання яких Вам потрібно застосувати на практиці, а програма допомагає їх швидше вивчити.

Програма полегшує та прискорює процес вивчення незнайомих слів, зберігає незнайомі слова їх транскрипцію та переклад у власну базу даних, спрощує пошук слів та групує їх відповідно до запитів користувача.

Методика даної програми розрахована на людей, які вже мають певний багаж знань. Користувач повинен володіти базовими знаннями граматики англійської мови [6].

Спеціалізація даної програми – вивчення слів. Наведемо приклад: учень, який вирішив досконало вивчати мову, придбав книжку англійською мовою. Він читає перший розділ і виписує незнайомі йому слова. Далі записує їх у «Словничок» і починає їх вивчати. Потім учень починає читати наступний розділ. Звичайно ж, він неодмінно зустрінеться із словами, що були незнайомі для нього в першому розділі. Таким чином з'являється практика використання слів. Прочитавши певну кількість розділів учень має змогу повторити всі нові слова з даної книжки, цим самим закріпити їх знання.

Читаючи книжки, дивлячись фільми або слухаючи пісні англійською мовою і при цьому працюючи зі «Словничком», можна швидко досягти бажаних результатів при поповненні свого активного словникового запасу.

Усі ці можливості дають структурні компоненти навчальної програми: функція вводу інформації, відображення груп в інтерфейсі, фільтри, створення та редагування списку гри, друк записів, нагадування про неперекладені слова.

Ввід інформації

База даних зберігається в типізованому файлі Data/DataBase.db. Його структура виглядає наступним чином:

```
TEnglish = record
```

```
    Number, TypeNumber:integer;
```

```
    Word, Transcript, Translate:string[255];
```

```
end;
```

де Number – порядковий номер запису;

TypeNumber – індекс групи;

Word, Transcript, Translate – відповідно слово, транскрипція , переклад.

Нові записи створюються за допомогою діалогового вікна «Словник». У ньому заповнюються всі поля запису, крім Number. Значення цьому полю надає програма. При роботі з файлом бази даних використовуються такі основні процедури:

```
procedure InitializeFile;
```

```
procedure
```

```
WriteFile(W,Trpt,Trlt:string;ToType:integer;position:integer=-1);
```

```
procedure BeginRead;
```

При завантаженні програми виконується процедура InitializeFile. Вона спочатку перевіряє його наявність. Якщо він відсутній, то підпрограма дає запит користувачу на його

створення. Далі, якщо файл знайдено, відбувається зчитування процедурою BeginRead.

Процедура

WriteFile(W,Trpt,Trlt:**string**;ToType:integer;position:integer=-1)

відповідає за створення та редагування записів. Перші чотири параметри редагує користувач. Це відповідно поля: «Слово», «Транскрипція», «Переклад», «Індекс групи». Параметр position вказує на дію, яку потрібно виконати. По замовчанню position=-1. Це означає, що потрібно створити новий запис. Якщо значення параметру більше мінус одиниці, то процедура замінює запис із індексом position на новий. Це використовується при редагуванні записів.

Для зручності користувача, при наданні фокуса на поле вводу «слово» чи «транскрипція» розкладка клавіатури змінюється на англійську. Якщо фокус отримує поле «переклад», то - на українську.

Для зручності пошуку слів та їх упорядкування створено систему групувань. По замовчанню стоїть група «невідома категорія». Користувач може створювати власні групи та редагувати їх зовнішність за допомогою редактора відображення груп.

Функція «вставити в поля» перевіряє буфер обміну і, якщо в ньому зберігається стрічка, в якій міститься слово, транскрипція і переклад, вставляє ці дані у відповідні поля. Цю функцію можна використовувати, коли, наприклад, потрібно ввести слова, які пишуться однаково, але читаються по-різному. Наприклад:

- 1) close [klou**z**] зачиняти; закінчити; кінець;
- 2) close [klou**s**] зачинений; відлюдний; задушливий; докладний; близько, поруч.

Дана програма передбачає можливість автоматичного вибору незаписаних слів із тексту. Ця функція реалізована в модулі uText та інкапсульована в клас TChooseWords.

Клас – це складна структура, яка включає, крім опису даних, опис процедур і функцій, які можуть бути виконані над представником класу – об’єктом. Дані класу (внутрішні змінні) називаються полями, процедури й функції – методами. Процес об’єднання змінних і методів, в результаті чого і з’являється клас, називається інкапсуляцією.

Далі буде розглянено принцип роботи алгоритму в загальних рисах, оскільки даний клас є одним із найоб’ємніших в програмі.

За допомогою **procedure Start(Texts:string)** вводиться текст перевірки, який далі зберігатиметься в полі класу **fText:string**. Далі відбувається знищення символів, які потрібно ігнорувати. Ця функція реалізована за допомогою чотирьох методів:

procedure ClearOut(Symbols:string);

procedure Replace(Symbols,ReplaceTo:string);

procedure ClearSelectedWaste;

procedure ClearArticles;

Методи **ClearOut(Symbols:string)** і **Replace(Symbols,ReplaceTo:string)** є основними. Інші дві працюють, використовуючи їх. Даний метод знищує з поля **fText** всі підстрічки, які рівні **Symbols**. При цьому береться до уваги те, що перед або після цієї підстрічки повинен бути пробіл (попередньо за допомогою методу **Prepare** виконується заміна символів переходу на нову строку на пробіл). Тобто символи не видаляються із середини слова.

Метод **ClearSelectedWaste** видаляє символи відповідно до списку ігнорованих символів, **ClearArticles** – артиклі.

Після очищення робочого тексту від зайвих символів відбувається формування списку незнайомих слів. Це виконує метод BreakToWords за таким алгоритмом: у масив fWords записуються слова, тобто підстрічка, яка знаходиться між пробілами. При цьому перевіряється наявність даного слова у масиві fWords, щоб не було повторювань. Але оскільки у словнику можуть зберігатися не лише слова, а і словосполучення, то попередньо створюється список існуючих словосполучень. За це відповідає метод SelectRichContain, який заносить у масив fRichContain вміст кожного поля «слово» із головного масиву work, що містить пробіл. Далі методом ClearOut(Symbols:string), де Symbols почергово дорівнює кожному з елементів масиву fRichContain, із робочого тексту fText видаляються існуючі словосполучення. Лише після цього виконується метод ClearArticles, оскільки словосполучення може містити артиклі та метод BreakToWords.

Після створення списку слів метод VerifyByWA перевіряє кожне знайдене слово із базою даних. У разі відсутності поточне слово записується у список незнайдених слів.

Відображення груп в інтерфейсі

Для того, щоб користувачу було наглядно зрозуміло, до якої групи належить будь-який запис, в програмі реалізовано модуль UGroupsOptions, що дозволяє налаштувати параметри відображення кожної із них. Для того щоб скористатися цією функцією, потрібно в головному меню обрати «Вигляд» - «Група». Відкриється редактор відображення груп. В ньому можливо відредагувати кожне поле групи за такими параметрами як стиль (курсив, підкреслений, напівжирний), колір надпису та фону, назва та величина шрифту.

Стили груп зберігаються у файлі Types\TypesStyle.tsf. Даний файл має такий формат запису:

TFontPar = record

Size:integer; // розмір шрифту

Name:string[255]; // назва шрифту

Color,bgColor:TColor; // колір шрифту, фону

Bold,Italic,UnderLine:boolean; // стилі

end;

Модуль UgroupsOptions складається з таких основних процедур:

procedure ReadStyleFile;

procedure writeStyleFile;

procedure InitializeStyleFile;

procedure CreateNewStyle;

procedure FillArray;

Процедура ReadStyleFile зчитує з файлу дані і заносить їх у масив TFontPar типу TFontPar. WriteStyleFile записує дані. При створенні першого запису у «Словничку» виконується процедура InitializeStyleFile, яка створює файл та заповнює його даними, які використовуються по замовчанню.

Підпрограма CreateNewStyle створює новий стиль. Вона викликається при створенні користувачем нової групи.

Процедура FillArray використовується при графічному виведенні даних у самому редакторі.

При перерисовці компонента відображення записів відбувається зміна елементів відображення відповідно до стилю групи, до якої належить поточний запис.

Фільтри

Система фільтрування складається з трьох фільтрів та системи, що відповідає за використання декількох фільтрів одночасно. Всі фільтри інкапсульовано в клас TFind. Ось основні процедури:

procedure FindFirst(First:string;ApplyToActiveFilter:boolean);

```
procedure Find(word:string;ApplyToActiveFilter:boolean);  
procedure ShowGroup(GroupIndex:integer;  
                    ApplyToActiveFilter:boolean;add:bool  
                    ean);
```

Процедура FindFirst здійснює пошук у полі «Слово» і виводить всі записи, дане поле якого розпочинається на параметр First.

Процедура Find здійснює пошук у полі «Переклад» і виводить всі записи, дане поле якого містить стрічку, яка аналогічна параметру word.

Процедура ShowGroup виводить всі записи, які відносяться до групи, індекс якої дорівнює параметру GroupIndex. Параметр add вказує, як потрібно вивести дані на екран. Якщо add=false, то процедура виведе на монітор лише дану групу. В іншому випадку дані будуть додані в кінець виведених записів. Дана функція використовується при запиті на фільтрацію декількох груп. Спочатку, при побудові списку першої обраної групи, add=false. Фільтрація наступних груп відбувається з add=true. Тобто спочатку виводиться перша група, а далі додаються записи, що належать наступній.

Всі щойно розглянені процедури мають однаковий параметр булінівського типу ApplyToActiveFilter. В залежності від нього, дані процедури змінюють область пошуку даних. Якщо ApplyToActiveFilter= false, то пошук даних відбувається по всій базі даних. В іншому випадку лише в виведених записах. Даний параметр використовується при роботі зі списком фільтрів, коли потрібно спочатку відфільтрувати дані за одним критерієм, а потім щойно виведені дані - за іншим.

Користувач може скористуватися можливістю використати більш ніж один фільтр за допомогою панелі «список фільтрів». Вона складається із списку фільтрів, які користувач може

вмикати і вимикати за бажанням, та кнопок, що додає останній використаний фільтр та видаляє виділений.

Система використання серії фільтрів працює так: у глобальній змінній `KindOfLastUsedFilter`: `TFilterKind` зберігається тип останньої фільтрації, де `TFilterKind = (fk_Group, fk_Word, fk_Translate, fk_None)`. У змінній `LastUsedFilter`: `string` зберігається умова фільтрації. При використанні фільтрації типу «слово» чи «переклад», тобто `KindOfLastUsedFilter := fk_Word` чи `fk_Translate`, у `LastUsedFilter` зберігається умова фільтрації у первісному вигляді. Якщо використовувався останній раз фільтр «група», то у ньому записано перелік індексів груп, взятий у трикутні дужки. Дані змінні використовує процедура `AddFilter`, яка додає останній використаний фільтр. Дані про кожен фільтр містяться в масиві `Filter` типу `TFilter`, де

TFilter = record

`Kind:TFilterKind;` // тип фільтрації

`Filter:string;` // умова

end;

При активації списку фільтрів чи їх зміни умови використовується процедура `ToFilter(Ordered:boolean)`, де `Ordered` показує чи потрібно фільтрувати в даний момент. Даний параметр був створений виключно для зручності написання подальшого коду.

Процедура `ToFilter` реалізована таким чином: якщо параметр `Ordered= true`, то відбувається фільтрація по принципу черги. Через цикл програма перевіряє чи відмічено поточний фільтр у списку. Якщо так, то в залежності від типу фільтра відбуваються наступні операції. Якщо `Filter[i].Kind` типу «слово» чи «Переклад», де `i` номер фільтру, то відбувається фільтрація за полем `Filter`. Якщо ж `Filter[i].Kind` типу «Група», то використовується функція `ExtractNext(var Text:string):integer`.

Вона із Text вибирає індекс першої групи, що і є результатом, і видаляє його з параметра. Якщо у стрічці не було знайдено індекса, то результати дорівнюють -1.

Друк записів

Якщо користувачу потрібно вивчити слова, але у нього немає можливості довго сидіти за комп'ютером, то зручно ці слова роздрукувати. Для цього було написано модуль, в якому реалізовано вивід обраних записів на друк. Для редагування сторінки використовують панель «редагування списку друку». Якщо вона відсутня, то потрібно в головному меню «Вигляд» відмітити назву цієї панелі.

Кожний запис містить в собі слово, його транскрипцію та переклад. Транскрипція обов'язково береться в квадратні дужки. Було помічено, що стрічку, яка містить запис, умовно можна поділити на три частини:

- 1) стрічка, що знаходиться попереду відкритої квадратної дужки (це слово);
- 2) стрічка, що знаходиться між квадратними дужками, включаючи їх (це транскрипція);
- 3) стрічка, яка знаходиться після закритої квадратної дужки (це переклад).

Зважаючи на це, ми вирішили створити інструменти, за допомогою яких можна було б редагувати окремі поля запису у списку друку. Панель «редагування списку друку» складається з чотирьох закладок: «слово», «транскрипція», «переклад», «загальна корекція». Щоб відредагувати на сторінці всі поля «слово», потрібно перейти на відповідну закладку панелі інструментів. В даній закладці можна змінити колір, шрифт, його розмір, стиль (підкреслений, курсив, напівжирний). Всі інші закладки, крім "Транскрипція" і «загальна корекція», мають такі ж інструменти редагування. У зв'язку з тим, що для транскрипції

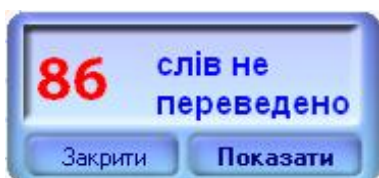
використовується власний шрифт, для поля "Транскрипція" відсутня можливість редагування шрифту. Інструменти закладки «загальна корекція» здійснюють редагування виділеної частини. У ній немає можливості редагувати шрифт. Тут знаходяться інструменти вирівнювання тексту.

Над сторінкою знаходяться інструменти, за допомогою яких виставляють відступи зліва, справа та абзац.

Щоб роздрукувати створену сторінку, потрібно в головному меню обрати "Список друку" - "роздрукувати". Після цього користувач матиме можливість вивчати слова без використання комп'ютера. Коли в нього з'явиться можливість працювати з "Словничком", користувач зможе перевірити та вдосконалити знання слів за допомогою гри.

Нагадування про неперекладені слова

Для того, щоб у «Словничку» знаходилися лише записи з перекладеними словами, у програмі діє нагадування про неперекладені слова (мал 1.1). Воно реалізоване у вигляді



Мал 1.1

невеличкого діалогового вікна, яке з'являється при першому завантаженні бази даних поточного сеансу та після виходу з режиму гри. У даному діалоговому вікні великими червоними літерами відображається кількість непереведених слів. У ньому є дві кнопки: «Закрити» - закриває дане вікно, «Показати» - виводить всі неперекладені слова.

Реалізація цієї функції досить проста. Спочатку перевіряються всі записи на наявність шуканих записів і записується їх кількість, якщо такі присутні. Натиснувши «Показати», використовується метод FindFirst класу TFind, де параметр First дорівнює «<<».

1.2 Гра, як структурний компонент навчальної програми «Словничок»

Не потрібно нікому доводити той факт, що гра не тільки розважає, дає можливість відпочити, але й навчає. Такий модуль в програмі як гра написати досить складно, тому що основною її функцією має бути інтерактивна співпраця з користувачем. Потрібно створювати гру, яка не тільки буде отримувати результати від користувача, але й буде опрацьовувати їх і виводити результати.

Створення та редагування списку гри

Для того, щоб розпочати вивчення слів за допомогою гри, потрібно спочатку створити список слів. Для цього було розроблено інструменти ручного створення списку і автоматичного. До ручних інструментів належать: «Додати поточний запис», «Видалити поточний запис», «Додати всі виведені записи», «Очистити список».

Ми помітили, що, зазвичай, незнайомі слова знаходяться в кінці «Словничка». Тому для більшої зручності формування списку було створено діалогове вікно "Налаштування гри". За допомогою нього користувач може скористатися автоматичним вибором останніх записів певної групи чи бази даних. Щоб вибір відбувався лише з певної групи, потрібно спочатку її позначити і відмітити "Застосувати параметри до групи". Якщо було відмічено декілька груп, то з кожної із них буде вибрано задану кількість записів (якщо задане число не більше за кількість записів, що належать до обраної групи). Вибір також враховує те, що запис може містити непереведене поле "переклад". Для того, щоб у процесі гри не виникало незручностей, дані записи ігноруються при автоматичному виборі.

Функція перенесення зміненого запису має велике значення для автовибору останніх слів. Змінений запис містить нову відредаговану інформацію. Це означає, що користувач може не володіти нею.

Якщо користувач вирішить випробувати свої знання, він може зробити це двома способами. Перший спосіб – це додати до списку гри всі записи. Він надає змогу перевірити всі слова. Але коли у "Словничку" знаходиться багато записів, то даний спосіб не практичний. Для того, щоб перевірити свої знання таким чином, потрібно багато часу. Другий спосіб – додати у список гри задану кількість записів, які вибрані абсолютно випадковою. Для цього у діалоговому вікні "Налаштування гри" потрібно перемикач переключити на "вибрати" і ввести кількість. При цьому можна враховувати лише окремі групи, як у попередній функції. Також є можливість вибрати певну кількість записів із кожної групи. Для цього потрібно поставити прапорець на «з кожної».

Функції автоматичного вибору інкапсульовані в клас TChoose. Ось його основні процедури:

```
procedure Last(Number:integer); // вибір останніх записів
```

```
procedure Randoms(Number:integer;overall:boolean=false); //  
випадковий
```

Саме ці процедури мають широке поле видимості **public**. Детально буде розглянуто лише побудову процедури Randoms, оскільки перша досить проста.

Randoms використовує процедури та змінні області приватної видимості **private**. Використовувати методи даної області можливо лише у даному модулі. Розглянемо ці процедури:

```
procedure SelectGroup(GroupIndex:integer);
```

```
procedure MixSelectedGroup;
```

procedure AddToEnum(Count:integer);

procedure MixEnum;

Процедура **SelectGroup(GroupIndex:integer)** заносить у інкапсульований масив **fSelectedGroup** типу **integer** ідентифікаційні номери записів, що належать до групи, індекс якої рівний **GroupIndex**.

Процедура **MixSelectedGroup** перемішує позиції даного масива. Це відбувається за таким алгоритмом: використовуючи цикл, для кожного елемента виконуються наступні дії:

- генерується методом **Random** випадкове число в межах довжини масиву;
- обмінюється поточний елемент масиву із елементом згенерованого числа.

Процедура **AddToEnum(Count:integer)** додає **Count** перших уже перемішаних елементів масиву **fSelectedGroup** до масиву **fEnumeration** того ж типу. Якщо користувач вибрав декілька груп і не відмітив прапорець «з кожної» (вибрати з кожної групи), то відбувається перемішування масиву **fEnumeration** процедурою **MixEnum**, яка аналогічна **MixSelectedGroup**. Після цього відповідно до параметра **Number**, опублікованої процедури **Randoms**, додається у список гри перші **Number** елементів масиву **fEnumeration**. Параметр **overall** вказує чи потрібно вибирати в сумі **Number** елементів **overall= true** чи з кожної групи **overall=false**.

Гра і її функції

Модуль, який відповідає за реалізацію гри, складається із таких головних функцій:

- 1) функція перемішування слів у поточному перекладі слова;
- 2) функція, що генерує послідовність виведення запитів;

Для повної функціональності програми, користувач повинен дотримуватися правил вводу інформації. Якщо всі вимоги були виконані, то доречно скористуватися функцією змішування слів. Вона працює так: спочатку переміщуються позиції груп слів, що розділені крапкою з комою. Потім у кожній із цих груп відбувається перемішування слів, що розділені комами. Таким чином дана функція перемішує слова, зберігаючи при цьому відповідність змісту кожної групи.

Функція перемішування слів реалізована п'ятьма процедурами:

```
procedure BreakApart(Symbol:string);  
procedure ClearSpaces;  
procedure Mix(mFrom,mTo:integer);  
procedure BreakInsideApart(Symbol1,Symbol2:string);  
procedure LetItMix(txt:string;var mixed:string).
```

Вони використовують спільні змінні:

```
fPartsCount:integer;// кількість частин  
fText :string; // текст, який потрібно розбити на частини  
fParts:Array of string;// масив, що містить розділені
```

частини

```
fMixedNumbers :array of integer; // масив, що містить  
    послідовність виводу розділених елементів, що  
    знаходяться в масиві fParts.
```

Процедура **BreakApart(Symbol:string)** розділяє стрічку **fText** на частини і заносить їх в масив **fPart**. Параметр **Symbol** – це стрічка-дільник, відповідно до якої буде відбуватися поділ на частини. Далі виконується процедура **ClearSpaces**. Вона видаляє всі останні пробіли, щоб потім не виникало проблем при формуванні нової послідовності виводу слів чи їх груп.

Процедура **Mix(mFrom,mTo:integer)** – генератор послідовності виводу. **mFrom** – мінімальне значення числа у

послідовності. В нашому випадку йому завжди відповідає одиниця. mTo – кінцеве. Йому надається значення, що відповідає кількості розділених частин. Згенерована послідовність чисел зберігається в масиві fMixedNumbers.

Процедура BreakInsideAparts(Symbol1,Symbol2:string), за допомогою вищеописаних процедур, формує нову послідовність груп та їх слів. Процедурою BreakAparts, де параметр Symbol:=Symbol2, заноситься в масив fPart групи слів. Далі дублюється масив fPart в локальний масив DParts. Після цього процедурою BreakAparts, де параметр Symbol:= Symbol1, кожен елемент DParts розбивається на окремі слова, що зберігаються в масиві fPart. За допомогою процедури Mix одержуємо послідовність слів, які потім записуються в той же самий елемент DParts. Такі дії відбуваються над кожним елементом даного масиву, після чого Mix'ом генерується нова послідовність розміщення груп слів.

Процедура LetItMix(txt:string;var mixed:string), де txt – текст-оригінал, mixed – стрічка, яка буде містити перемішаний варіант тексту, узагальнює всі вищеописані процедури і використовується як основна.

procedure LetItMix(txt:string;var mixed:string);

begin

fText:=txt;// ініціалізація

if Copy(fText,length(fText),1)<>' ' **then** // перевірка на коректне закінчення

fText:=fText+' ';

BreakInsideAparts(' ',' '); // одержання нової послідовності

mixed:=fText;// вивід перемішаного тексту

end;

Ідея створення цієї функції прийшла після певного терміну користування програмою. Було помічено, що коли слово має не

один переклад, а декілька, то, зазвичай, запам'ятовуються лише перші варіанти перекладу. Просто користувач, побачивши на початку знайомі йому слова й не дочитавши весь їх перелік до кінця, відразу вводить переклад. Щоб уникнути цього явища, була створена функція перемішування. Практика її використання показала, що якість запам'ятовування слів значно збільшилась.

За вивід запитів відповідає функція генерування послідовності слів. Вона виводить для перевірки одне із вибраних слів. Якщо відповідь, дана користувачем, хибна, то кількість виведення поточного слова збільшується на одиницю, якщо правильна, то зменшується. Тобто, якщо користувач помилиться N раз, то йому потрібно ввести $N+1$ правильних відповідей, щоб даний запит більше не виводився в поточній грі.

Дана функція складається з процедур:

procedure Initialize;

procedure RandomWordExciption;

procedure TakeTranslate;

procedure TfGame.eTranslateKeyPress(Sender: TObject; **var** Key: Char);

procedure TfGame.Timer1Timer(Sender: TObject);

Процедура Initialize ініціює всі потрібні для гри значення. Вона використовується при старті гри. RandomWordExciption обирає слово для запиту. Перший вибір може впасти на будь-яке слово. Користувач вводить переклад і натискає "Enter", що викликає обробку події eTranslateKeyPress. Ця процедура робить поле вводу неактивним і перевіряє правильність відповіді процедурою TakeTranslate. Відповідно до налаштування і даної відповіді задається властивість Timer1.interval і активується таймер Timer1. TfGame. Timer1Timer активує поле вводу і за допомогою RandomWordExciption виводиться наступний запит.

Процедура TakeTranslate при правильній відповіді зменшує кількість звертань до цього запису на одиницю, в іншому випадку збільшує на одиницю.

Для полегшення вивчення слів у користувача є можливість налаштувати генератор, щоб він підряд повторював задану кількість разів той запит, на який була дана неправильна відповідь. Щоб скористатися даною функцією, потрібно викликати діалогове вікно «Налаштування гри» і в полі «К-ть контрольних повторів при помилці» ввести значення повторів.

У грі можна також налаштувати такі параметри: час показу правильного варіанта при наданні хибної відповіді і при наданні правильної відповіді. Для цього потрібно викликати діалогове вікно "налаштування гри" і змінити дані в області "Час показу відповідей".

Всі дані, що не стосуються списку гри, можна редагувати під час її проходження. Кнопка "Параметри" відповідає саме за це. Вона викликає діалогове вікно «Налаштування гри», в якому доступні лише вищесказані функції.

В цьому розділі розглянуто основні функції і елементи однієї із складових програми «Словничок» - гри. На нашу думку впровадження такого елемента в програму сприятиме кращому запам'ятовуванню і зацікавленню учнів програмою.

РОЗДІЛ II СПЕКТР ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ ПРОГРАМИ «СЛОВНИЧОК»

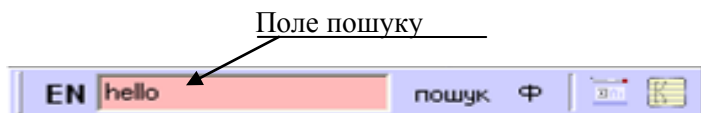
2.1. Основні функціональні можливості створеної програми

При створенні будь-якої програми досить складно досягнути такого результату, щоб вона була не тільки функціональна, грамотно написана, а й відповідно оформлена. Оскільки простий і зрозумілий інтерфейс створює умови для зручного і легкого користування програмою, тому досить важливо написати програму, яка буде мати не тільки широкі функціональні можливості, а й простий і доступний вигляд і управління.

Програма «Словничок» надає можливість вводити інформацію за допомогою трьох способів. Для того, щоб ввести одне слово, потрібно:

Спосіб 1. В головному меню вибрати «Слово» - «Додати нове» або викликати контекстне меню, нажавши праву кнопку мишки на таблиці виведення записів і вибрати «Додати нове».

Спосіб 2. У поле пошуку ввести слово, у даному випадку «hello» (мал. 2.1), і натиснути «insert».

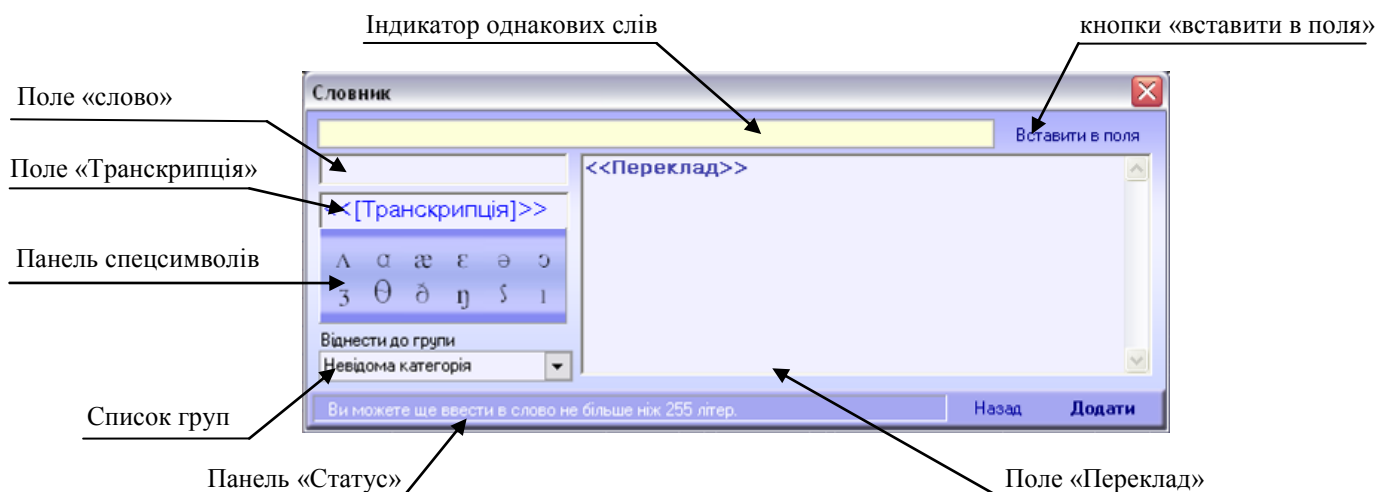


Мал. 2.1

Незалежно яким способом користувач скористався, програма виведе діалогове вікно «Словник». За його допомогою відбувається занесення даних у базу записів. Дане діалогове вікно складається з трьох полів вводу («Слово», «Транскрипція»,

«Переклад»), списку груп, панелі спецсимволів, індикатора однакових слів та кнопки «вставити в поля». Внизу знаходиться панель статусу, яка показує кількість символів, можливих для вводу в поточне поле (мал. 2.2).

Якщо користувач скористався першим способом вводу, то з'явиться вищесказане діалогове вікно з незаповненими полями. Зазвичай, спочатку заповнюють поле «слово». У тому випадку, коли поточне слово є у базі даних, то індикатор однакових слів виводить його разом з перекладом. Ця функція є корисною, коли



Мал. 2.2

користувач забувся, що дане слово було занесено раніше. Вона запобігає випадковому дублюванню записів із однаковими словами. Але дана функція лише попереджає про ідентичність слів і не забороняє додавати поточний запис у базу даних. Це зумовлено тим, що в англійській мові є слова, які пишуться однаково, а читаються по-різному.

Заповнюючи поле «транскрипція», користувач може скористатися панеллю спецсимволів. Вона дозволяє вставити обраний символ в поточну позицію даного поля вводу. Спецсимвол можна також ввести за допомогою клавіатури. Для цього порібно натиснути декілька разів на клавішу, символ якої

схожий на потрібний спецсимвол. Це такі клавіші як: «а», «е», «о», «і», «п», «s».

Введення перекладу потребує дотримання наступних правил:

- 1) щоб розділити слова схожого змісту, використовується кома;
- 2) щоб розділити слова чи групи слів з різним змістом, використовується крапка з комою. Забороняється їх розділяти за допомогою чисел з дужками. Наприклад: 1)...2)...;
- 3) використання дужок можливе лише в тому випадку, коли потрібно щось уточнити. Відкрита дужка обов'язково повинна бути закритою;
- 4) переклад не повинен розпочинатися двома відкритими трикутними дужками "<<". Всі слова, переклад яких так розпочинається, вважаються неперекладеними. При цьому програма в деяких випадках працюватиме некоректно.

Для зручності пошуку слів та їх упорядкування, створено систему групувань. По замовчанню стоїть група «невідома категорія». Для створення нової групи потрібно ввести нову назву в поле вводу списку і натиснути «Enter». Щоб перейменувати групу, потрібно в контекстному меню вибрати «Перейменувати». З'явиться діалогове вікно, в якому потрібно ввести нову назву і натиснути кнопку «Добре».

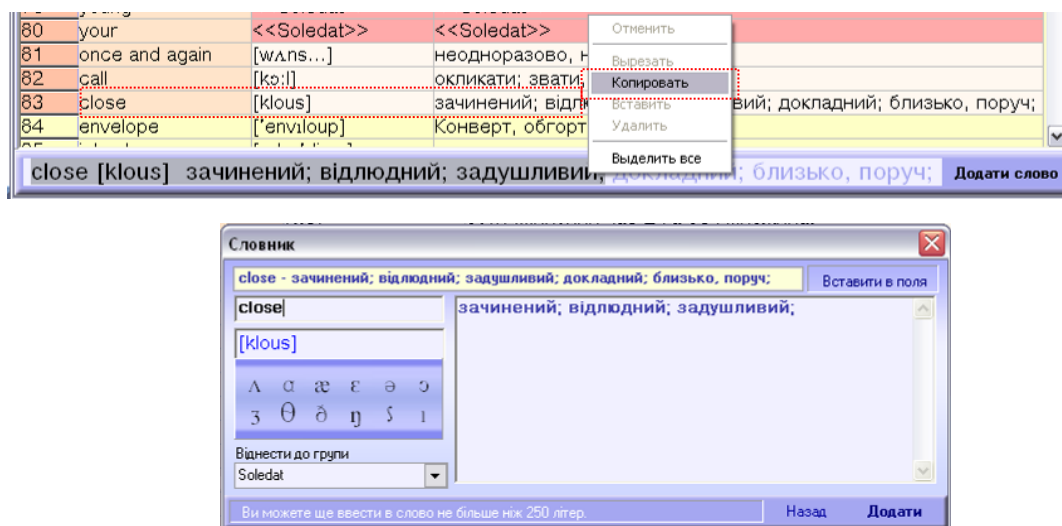
Після того, як користувач заповнив всі поля, щоб додати запис у базу даних, потрібно натиснути «Додати». При наступному виклику цього діалогового вікна назва групи автоматично виставляється на ту, яка була обрана минулого разу. Це досить зручно, коли користувач вводить серію слів, які відносяться до однієї групи.

Функція «вставити в поля» перевіряє буфер обміну і, якщо в ньому зберігається стрічка, в якій міститься слово, транскрипція і переклад, вставляє ці дані у відповідні поля. Цю функцію можна використовувати, коли, наприклад, потрібно ввести слова, які пишуться однаково, але читаються по-різному. Наприклад:

3) close [klouz] зачиняти; закінчити; кінець;

4) close [klous] зачинений; відлюдний; задушливий; докладний; близько, поруч.

Спочатку заносимо у базу даних перший запис. Далі виділяємо його в таблиці відображення і із панелі статусу головного вікна копіюємо стрічку. Викликаємо діалогове вікно «Словник» і натискаємо «Вставити в поля». Після цього всі поля заповняться і залишиться лише зробити деякі зміни.



Мал. 2.3

Поговоримо про другий спосіб введення слова у словник. Даний спосіб, оскільки слово вводиться в полі пошуку, дозволяє ще до відкриття діалогового вікна «Словник» побачити чи є дане слово у словнику. Якщо воно відсутнє, то поле вводу забарвиться в червонуватий колір, якщо ні, то у таблиці відображення записів виведуться всі записи, слова яких починаються на введену в поле пошуку стрічку. Натиснувши «insert» з'явиться діалогове вікно «Словник», в якому поле «Слово» буде заповнене відповідно до

поля пошуку, а фокус вводу буде мати поле «транскрипція». Далі всі операції відбуваються аналогічно до першого способу.

Якщо користувачу потрібно вибрати незнайомі слова із тексту в електронному вигляді, то доречно скористатися функцією автоматичного вибору слів. Для цього потрібно в головному меню обрати «Фільтр»-«Вибрати незнайомі слова». З'явиться однойменне діалогове вікно. Воно складається з поля вводу тексту і панелі налагодження. В поле вводу тексту вставляють потрібний текст. Панель налаштування складається з двох частин: параметри фільтру та параметри запису слів. За допомогою «параметри фільтру» можна зробити наступні дії: вибрати, до якої групи віднести вибрані слова; налаштувати список символів, які фільтр буде ігнорувати; налаштувати список можливих закінчень, які будуть враховані під час вибору слів; включити чи виключити функцію, що не враховує артиклі. Зробивши всі налаштування, потрібно натиснути кнопку «Вибрати». Після завершення операції, в правій частині панелі налагодження з'явиться перелік вибраних слів, які пропонується занести в словник. Користувач може виключити з цього списку будь-яке слово, знявши ярличок. Щоб занести всі відмічені слова в словник, потрібно натиснути кнопку «Додати». Після цього в базу даних буде додано записи, слова яких відповідають вибраним, а транскрипція і переклад заповнений назвою обраної групи, взятої в подвійні трикутні дужки.

В програмі також існує можливість редагувати записи. Для цього потрібно використати один із способів виклику діалогового вікна редагування. Найпростіше це зробити подвійним клацанням мишки на потрібному запису. Можна також скористатися головним меню "Слово" – "Виправити", контекстним меню або клавішею швидкого реагування "F12". Будь-який із цих способів викликає вже знайоме нам діалогове вікно "Словник". У ньому

користувач може змінити дані даного запису. При цьому відредагований запис переміщується в кінець бази даних.

Фільтри

Для пошуку і фільтрації записів використовується три функції: пошук по в полі "слово", "переклад", фільтрація за групами.

Пошук в полі "Слово" використовується для того, щоб знайти потрібне слово в англійській мові. Ця функція виводить всі записи, поле "Слово" яких починається на введену стрічку для пошуку. Щоб скористатися цим фільтром, потрібно у полі пошуку на панелі "Фільтр" ввести потрібну стрічку. Якщо у базі даних є записи, що задовольняють фільтр, то програма їх виводить, якщо ні, то поле пошуку забарвлюється в червонуватий колір, і програма виводить всі записи. До речі, біля цього поля знаходиться кнопка-перемикач, завдяки якій при наданні фокусу полю мова вводу змінюється на обрану, якщо функція активна. Щоб змінити значення перемикача, потрібно натиснути на ньому або скористатися контекстним меню.

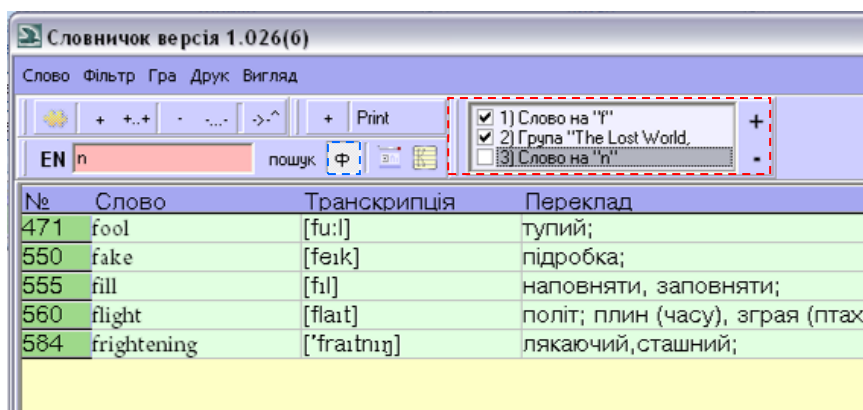
Пошук у полі "Переклад" використовується, щоб знайти слова в англійській мові, переклад яких відповідає введеному значенню. Ця функція виводить всі записи, "переклад" яких містить задану стрічку. Щоб скористатися цією функцією, потрібно ввести шукану стрічку в поле пошуку і натиснути "пошук" або клавішу "Enter".

Фільтрація за групами використовується, щоб вивести лише записи обраних груп. Для цього потрібно у головному меню "Фільтр" обрати "Вивести групи" або скористатися однойменною панеллю. Відкриється діалогове вікно "Фільтрувати за групами". У ньому міститься список усіх груп, в якому користувач повинен відмітити потрібні йому групи. Після цього натискають "Фільтрувати". Програма виведе всі відфільтровані записи. Щоб

вивести всі записи, потрібно в списку груп діалогового вікна "Фільтрувати за групами" обрати "Всі групи" або змінити фокус на поле пошуку.

Незалежно яким фільтром користувач скористався, на панелі статусу головної форми виведеться кількість відфільтрованих записів.

Дана програма також надає можливість використовувати декілька фільтрів одночасно. Щоб скористатися нею, потрібно спочатку її активувати. Для цього на панелі "Фільтр" потрібно включити перемикач «використовувати список фільтрів» (на ньому зображено «Ф», на мал. 2.4 обведено синьою лінією).



Мал.2.4

Панель «Список фільтрів» (на мал. 2.4 обведено червоною лінією) складається з: кнопок «Додати останній використаний фільтр»; списка відображення обраних фільтрів. Назва елементів списку формується таким чином: спочатку пишеться назва фільтру, потім у лапках записана умова. У тому випадку, коли використовувався тип фільтру «Група», то назви груп пишуться через кому.

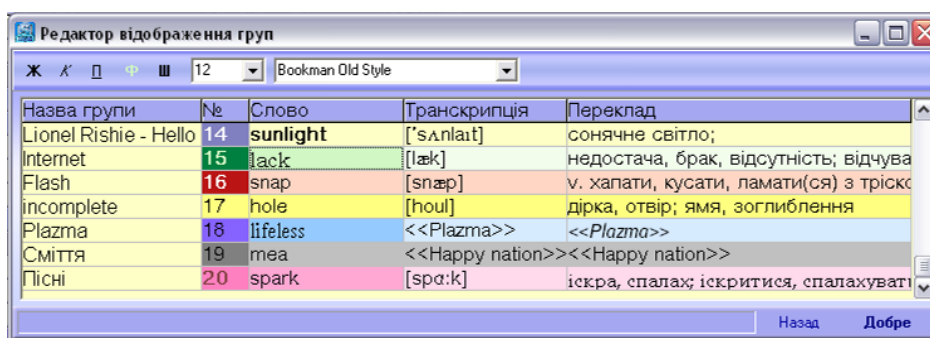
Кожний фільтр можна вимкнути, знявши прапорець біля нього.

Фільтрування відбувається по чергово з верху списку до низу. Якщо певний фільтр не дав результату, то він ігнорується. Наприклад, якщо у список занести по чергово різні групи, то

відображатиметься лише найближча до початку відмічена група. Те ж саме відбувається з використанням типу фільтрації «Слово».

Відображення груп в інтерфейсі

Щоб користувачеві було наглядно видно, до якої групи належить певний запис, дана програма надає можливість редагувати відображення груп. Для того щоб скористатися цією функцією, потрібно в головному меню обрати «Вигляд» - «Група». Відкриється редактор відображення груп (мал.2.5). В ньому можливо відредагувати кожне поле групи за такими параметрами як стиль (курсив, підкреслений, напівжирний), колір надпису та фону, назва та величина шрифту.



Мал. 2.5

Дане діалогове вікно складається з панелі редагування та списку груп у вигляді таблиці. Для того щоб внести зміни до відображення потрібного поля обраної групи, потрібно виділити відповідну комірку. Після цього вона редагується. Кнопки «Ф» і «Ш» викликають стандартне діалогове вікно вибору кольору. «Ф» відповідає за колір фону, «Ш» – шрифту.

Друк записів

Якщо Вам потрібно вивчити велику кількість слів, то доречно їх роздрукувати. Це надає можливість продовжувати їх вивчення без використання комп'ютера. Для редагування сторінки використовують панель «редагування списку друку».

Якщо вона відсутня, то потрібно в головному меню «Вигляд» відмітити назву цієї панелі.

Панель «редагування списку друку» складається з чотирьох закладок: «слово», «транскрипція», «переклад», «загальна корекція». Щоб відредагувати на сторінці всі поля "слово", потрібно перейти на відповідну закладку панелі інструментів. В даній закладці можна змінити колір, шрифт, його розмір, стиль (підкреслений, курсив, напівжирний). Всі інші закладки, крім "Транскрипція" і «загальна корекція», мають такі ж інструменти редагування. У зв'язку з тим що для транскрипції використовується власний шрифт, для поля "Транскрипція" відсутня можливість редагування шрифту. Інструменти закладки «загальна корекція» здійснюють редагування виділеної частини. У ній немає можливості редагувати шрифт. Тут знаходяться інструменти вирівнювання тексту.

Над сторінкою знаходяться інструменти, за допомогою яких виставляють відступи зліва, справа та абзац.

Щоб роздрукувати створену сторінку, потрібно в головному меню обрати "Список друку" - " роздрукувати". Після цього користувач матиме можливість вивчати слова без використання комп'ютера. Коли в нього з'явиться можливість працювати з "Словничком", користувач зможе перевірити та вдосконалити знання слів за допомогою гри.

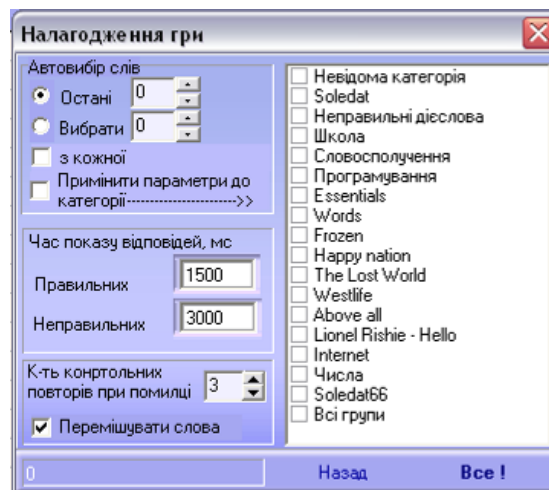
2.2 Інтерфейс гри навчальної програми «Словничок»

Створення та редагування списку гри

Як вже раніше згадувалось, для того щоб розпочати вивчення слів за допомогою гри, потрібно спочатку створити список слів. Відповідно до нього програма формуватиме запити. "Словничок" має декілька інструментів, за допомогою яких

можна створити список. Щоб додати виділений запис в список гри, потрібно на панелі гри натиснути кнопку "Додати поточне слово" або у контекстному меню вибрати "Додати до гри". Після цього у правій частині головної форми з'явиться список, який містить обраний запис. Використовуючи такий метод, можна додати будь-які записи. Для того, щоб знищити непотрібний запис із списку гри, потрібно його виділити і натиснути кнопку "Видалити поточний запис" на панелі інструментів гри або натиснути "Delete". Щоб видалити всі записи, потрібно натиснути кнопку "Очистити список" або скористатися комбінацією "Ctrl"+"Delete".

На панелі інструментів "гра" також знаходиться кнопка "Додати всі виведені записи". На перший погляд, дана функція не є практичною для використання. Але це не так. За допомогою використання різних фільтрів та їх списку можна по-різному налаштувати виведення даних (мал 2.6).



Мал. 2.6

Щоб скористатися автоматичним вибором записів, потрібно на панелі «Гра» натиснути «Налаштування гри» або скористатися клавішею «F5». З'явиться одноіменне вікно.

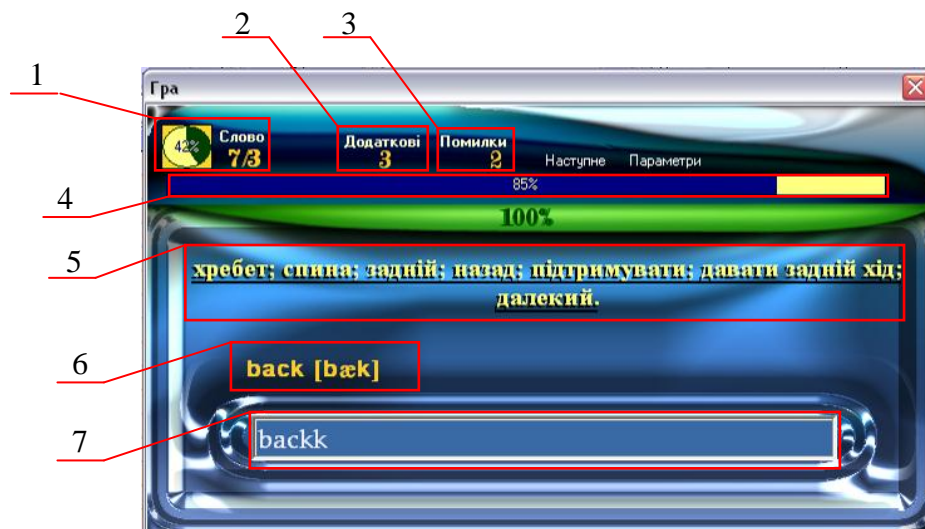
Дане діалогове вікно містить в собі панель «Автовибір слів». За її допомогою користувач може вибрати задану кількість останніх слів або випадкових. При цьому можна використати

лише певну групу, поставивши прапорець біля «Застосувати параметри до групи» і відмітивши назву групи із списку, що знаходиться справа. Якщо використовується випадковий вибір, то можна вказати додатковий параметр «з кожної». Якщо він увімкнений, то буде вибрано з кожної задану кількість слів.

Гра і її функції

Склавши список слів для гри, користувач може приступити до вивчення слів, натиснувши на панелі гри кнопку «Розпочати гру» або скориставшись клавішею швидкого реагування «F8». З'явиться вікно «Гра» (мал. 2.7)

Дане вікно складається з: прогрес гри (1); чисел, що показує кількість додаткових слів (2) і кількість помилок (3); кнопки «Наступне», яка виводить наступний запит; кнопки «Параметри», що викликає діалогове вікно «Налаштування гри», в якому можливо змінити налаштування, що стосуються лише до гри; індикатор рівня знань (4); запиту (5); показу правильної відповіді (6); поле введення відповіді (7).



Мал. 2.7

За допомогою діалогового вікна «Налаштування гри», можна налаштувати такі властивості як повтор слів, перемішування, час показу відповідей.

Гра відбувається таким чином: спочатку виводиться запит, далі користувач вводить відповідь і натискає «Enter». Відбувається перевірка відповіді. Якщо правильна, то виводиться транскрипція слова світло- фіолетового кольору. В іншому випадку правильний варіант відповіді помаранчевого кольору з транскрипцією. Далі відповідно до налаштування гри відбувається повтор слів. Ця функція створена для кращого запам'ятовування слів.

При закінченні гри виведеться запит, в якому запропонується зіграти ще. За бажанням користувач може натиснути «Так» або «Ні».

Взагалі використання такого розділу як гра є доцільним при вивченні не тільки нового матеріалу, але й для ефективнішого засвоєння матеріалу, який був вивчений раніше. Також доцільно використовувати гру при повторенні матеріалу, адже вона зацікавлює, викликає інтерес до того чи пам'ятає користувач матеріал, який давно пройшов.

Гра спонукає користувача виконати її завдання з максимальним числом даних про успіхи виконання [15].

На нашу думку, програма в цілому і зокрема гра буде допомагати учням і студентам при систематичному та свідомому вивченні іноземної мови. Необхідно підкреслити, що програма розрахована на самостійну роботу з нею, тому її користувач має бути позитивно мотивованим.

Список використаної літератури

1. Александровский А.Д. Delphi 4. Шаг в будущее. - М.: "ДМК", 1999 г.
2. Александровский А.Д. Технология программирования.- Рыбинск: РГАТА, 1999 г.
3. Архангельский А.Я. Программирование в Delphi. Учебник по классическим версиям Delphi Бином-Пресс, 2008. - 1158 с.
4. Бакнелл Д. Фундаментальные алгоритмы и структуры данных в Delphi - ДиаСофтЮП, 2003. – 560 с.
5. Берн Э. Игры, в которые играют люди. Люди, которые играют в игры: //М.: Прогресс, 1992. – 399 с.
6. Газман О.С. Роль игры в формировании личности школьника // Совет. педагогика, 1982. – № 9
7. Энго Ф. Самоучитель DELPHI 3. - Киев, 1998 г.
8. Фаронов В. Система программирования Delphi. – СПб.: БХВ-Петербург, 2006. – 191 с.
9. Культин Н.Б. Delphi 6 программирование на Object Pascal. – СПб.: БХВ-Петербург, 2001. – 237 с.
10. Культин Н.Б. Программирование в TurboPascal 7.0 и Delphi. - Санкт-Петербург, 1999 г.
11. Культин Н.Б. Основы программирования в Delphi 8 для Microsoft.NET Framework. Самоучитель - БХВ-Петербург, 2004. – 400 с.
12. Лисков Б. Дж.Гагер. Использование абстракций и спецификаций при разработке программ. М.: Мир, 1989 г.
13. Любавин С.А. Програмуємо в Turbo Delphi - НТПресс, 2008. - 320 с.
14. Основи нових інформаційних технологій навчання: Посібник для вчителів /Авт.кол.; За ред. Ю.І. Машбиця / Інститут технології ім. Г.С. Костюка АПН України. – К.: ІЗМН, 1997.

15. Перкас С.В. Ролевые игры на уроках английского языка. //Иностранные языки в школе. № 4, 1999г.
16. Пестриков В.М. Delphi на примерах - БХВ-Петербург, 2005. – 496 с.
17. Ревич Ю.В. Нестандартные приемы программирования на DELPHI - БХВ-Петербург, 2005. – 560 с.
18. Сухарев М.В. Основы Delphi профессиональный подход. – СПб.: НиТ, 2004, - 90 с.
19. Сухарев М.В. Основы Delphi. Профессиональный подход - Наука и техника, 2004. – 600 с.
20. Сорокин А.В. Delphi. Разработка баз данных - Питер, 2005. – 480 с.
21. Фленов М.Е. Библия Delphi - БХВ-Петербург, 2005. – 880 с.
22. Хендерсон К. Руководство разработчика баз данных в DELPHI 2.- Киев, 1996 г.
23. Шупрута В.В. Delphi 2005. Учимся программировать - ИТ Пресс, 2005. – 352 с.
24. <http://www.xserver.ru/computer/sredaprogr/bd/1/>
25. <http://www.nmarket.ru/program/delphi/lessons-2/>
26. <http://256bit.ru/Delphi7/>
27. Форум на www.delphimaster.ru