НТУУ «КПИ»

ФИВТ

Кафедра ВТ

Лабораторная работа №5

Выполнил:

студент группы ИВ-34

Мозговой Иван

Проверил:

Невдащенко М. В.

Номер зачетной книжки:

3416

Киев 2014г.

Определение задания:

Модифицировать лабораторную работу №3 таким образом: для букв, слов, предложений, знаков препинания создать отдельный классы. Слово должно состоять из массива букв, предложение из массива слов и знаков препинания, текст должен состоять из массива предложений

Описание программы:

Данная программа состоит из 8 классов, каждый из которых предназначен для хранения знаков препинания, букв, слов, предложений и текста целиком. Назначения каждого класса будут описаны в коде.

Код программы:

**package** kpi.oop.lab;

**import** java.util.ArrayList;

/\*\*

\*

\* **@version** 1.1 29 May 2014

\* **@author** Mozgoviy Ivan

\*

\*/

/\*\*

\* Это исполнительный класс.

\*

\*/

**public** **class** Main {

/\*\*

\*метод который создает текст.

\*

\*/

**public** **static** MyString del(MyString ms) {

Sentence[] s = ms.getSentences();

**for** (**int** i = 0; i < s.length; i++) {

ElBS[] w = s[i].getElBS();

**for** (**int** j = 0; j < w.length; j++) {

**if** (w[j] **instanceof** Word) {

Word wr = (Word) w[j];

Letter[] l = wr.toArrayLetter();

Letter ll = l[l.length -1];

ArrayList<Letter> al = **new** ArrayList<Letter>();

**for** (**int** k = 0; k < l.length; k++) {

al.add(l[k]);

}

**for** (**int** k = 0; k < al.size() - 1; k++) {

**if** (al.get(k).equals(ll)) {

al.remove(k);

k--;

}

}

l = **new** Letter[al.size()];

**for** (**int** k = 0; k < l.length; k++) {

l[k] = al.get(k);

}

w[j] = **new** Word(l);

}

}

}

**return** **new** MyString(s);

}

/\*\*

\* метод для ввода и вывода готового текста.

\*

\*/

**public** **static** **void** main(String[] args) {

MyString ms = **new** MyString(

"Qsdasdhadhbhqq sdfhsdbfjhwesdfeye! dsaf asdffffseff. asd sfyebf efsydfb wefyfydfd df dfdfdfdf..."

.toCharArray());

ms = *del*(ms);

System.*out*.println(ms);

}

}

**package** kpi.oop.lab;

/\*\*

\*

\* **@author** admin

\* абстрактный класс для хранения разделителей слов в одном массиве.

\*

\*/

**public** **abstract** **class** ElBS {

**public** **abstract** String toString();

}

**package** kpi.oop.lab;

/\*\*

\*

\* **@author** Ivan

\* класс, хранящий буквы.

\*

\*/

**public** **class** Letter {

**char** c;

**public** Letter(**char** ch) {

c = ch;

}

**public** **void** setChar(**char** ch) {

c = ch;

}

**public** **char** getChar() {

**return** c;

}

**public** String toString() {

**return** "" + c;

}

/\*\*

\* метод для перевода большой буквы в маленькую.

\*/

**public** **void** toLowerCase() {

**if** ((c >= 'A') && (c <='Z')) {

c += 'a' - 'A';

}

}

/\*\*

\* метод для сравнения букв.

\*/

**public** **boolean** equals(Object o) {

**if** (o **instanceof** Letter) {

Letter l = (Letter) o;

Letter m = **this**;

l.toLowerCase();

m.toLowerCase();

**if** (l.c == m.c){

**return** **true**;

} **else** {

**return** **false**;

}

} **else** {

**return** **false**;

}

}

}

**package** kpi.oop.lab;

**import** java.util.Arrays;

/\*\*

\*

\* **@author** Ivan

\* класс для "сборки" строки из массива слов и предложений.

\*

\*/

**public** **class** MyString {

**private** Sentence[] s;

**private** **boolean** isLetter(**char** c) {

**return** ((c >= 'A') && (c <= 'Z') || (c >= 'a') && (c <= 'z'));

}

**public** MyString(Sentence[] ss) {

s = ss;

}

**public** MyString(**char**[] c) {

**boolean** fl = **false**;

**char**[] a = **new** **char**[0];

**for** (**int** i = 0; i < c.length; i++) {

**if** ((c[i] != ' ') && (c[i] != '\t')) {

fl = **true**;

}

**if** (fl) {

**if** (c[i] == '\t') {

a = Arrays.*copyOf*(a, a.length + 1);

a[a.length - 1] = ' ';

} **else** {

a = Arrays.*copyOf*(a, a.length + 1);

a[a.length - 1] = c[i];

}

}

**if** ((c[i] == ' ') || (c[i] == '\t')) {

fl = **false**;

}

}

ElBS[] e = **new** ElBS[0];

**int** sW = 0;

**int** fW = 0;

fl = **false**;

**boolean** pr = **false**;

**for** (**int** i = 0; i < a.length; i++) {

**if** (isLetter(a[i])) {

**if** (!fl) {

fl = **true**;

pr = **false**;

sW = i;

}

} **else** {

**if** (!pr) {

fl = **false**;

fW = i;

Letter[] w = **new** Letter[0];

**for** (**int** j = sW; j < fW; j++) {

w = (Letter[]) Arrays.*copyOf*(w, w.length + 1);

w[w.length - 1] = **new** Letter(a[j]);

}

e = (ElBS[]) Arrays.*copyOf*(e, e.length + 2);

e[e.length - 2] = **new** Word(w);

e[e.length - 1] = **new** Separator(a[i]);

pr = **true**;

} **else** {

e = (ElBS[]) Arrays.*copyOf*(e, e.length + 1);

e[e.length - 1] = **new** Separator(a[i]);

}

}

}

Sentence[] sen = **new** Sentence[0];

**int** fS = -1;

**int** sS = 0;

**for** (**int** i = 0; i < e.length; i++) {

**if** (e[i] **instanceof** Separator) {

Separator sep = (Separator) e[i];

**if** ((sep.getChar() == '.') || (sep.getChar() == '!')

|| (sep.getChar() == '?')) {

sS = fS + 1;

fS = i;

ElBS[] aa = **new** ElBS[0];

**for** (**int** j = sS; j < fS + 1; j++) {

aa = (ElBS[]) Arrays.*copyOf*(aa, aa.length + 1);

aa[aa.length - 1] = e[j];

}

sen = (Sentence[]) Arrays.*copyOf*(sen, sen.length + 1);

sen[sen.length - 1] = **new** Sentence(aa);

}

}

}

s = sen;

}

**public** Sentence[] getSentences() {

**return** s;

}

**public** String toString() {

String res = "";

**for** (**int** i = 0; i < s.length; i++) {

res += s[i].toString();

}

**return** res;

}

}

**package** kpi.oop.lab;

/\*\*

\*

\* **@author** Ivan

\* Класс для предложений, состоящий из массивов слов.

\*

\*/

**public** **class** Sentence {

**private** ElBS[] e;

**public** Sentence(ElBS[] elbs) {

e = elbs;

}

**public** ElBS[] getElBS() {

**return** e;

}

**public** String toString() {

String res = "";

**for** (**int** i = 0; i < e.length; i++) {

res += e[i].toString();

}

**return** res;

}

}

**package** kpi.oop.lab;

/\*\*

\*

\* **@author** Ivan

\* класс для хранения букв и слов

\*

\*/

**public** **class** Separator **extends** ElBS {

**char** c;

**public** Separator (**char** ch) {

c = ch;

}

**public** **void** setChar(**char** ch) {

c = ch;

}

**public** **char** getChar() {

**return** c;

}

**public** String toString() {

**return** "" + c;

}

}

**package** kpi.oop.lab;

/\*\*

\*

\* **@author** admin

\* класс для слов, состоящих из массива букв.

\*

\*/

**public** **class** Word **extends** ElBS {

**private** Letter[] l;

**public** Word(Letter[] lt) {

l = lt;

}

**public** Letter getLetter(**int** i) {

**return** l[i];

}

**public** Letter[] toArrayLetter() {

**return** l;

}

**public** String toString() {

String res = "";

**for** (**int** i = 0; i < l.length; i++) {

res += l[i].getChar();

}

**return** res;

}

}

**package** kpi.oop.lab;

/\*\*

\*

\* **@author** Ivan

\* класс для знаков препинания.

\*

\*/

**public** **class** Punctuation {

**private** **char** p[] = { '.', ',', '!', '?', ';', ',', '-', ':' };

}

Выводы:

Данная лабораторная работа демонстрирует работу с классами. Также показывает возможности стандартных средств Java. В этой лабораторной работе были созданы 8 классов, каждый из которых выполнял свою определенную функцию. Было использовано наследование классов. Был использовано родительский абстрактный класс для хранения разделителей текста.