|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Робототехника и комплексная автоматизация (РК)

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**по курсу объектно-ориентированное программирование**

Студент Камалов Антон Павлович

Группа РК6-25Б

Тип задания Лабораторная работа №6

Вариант L38

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Камалов А.П.**

*подпись, дата фамилия, и.о.*

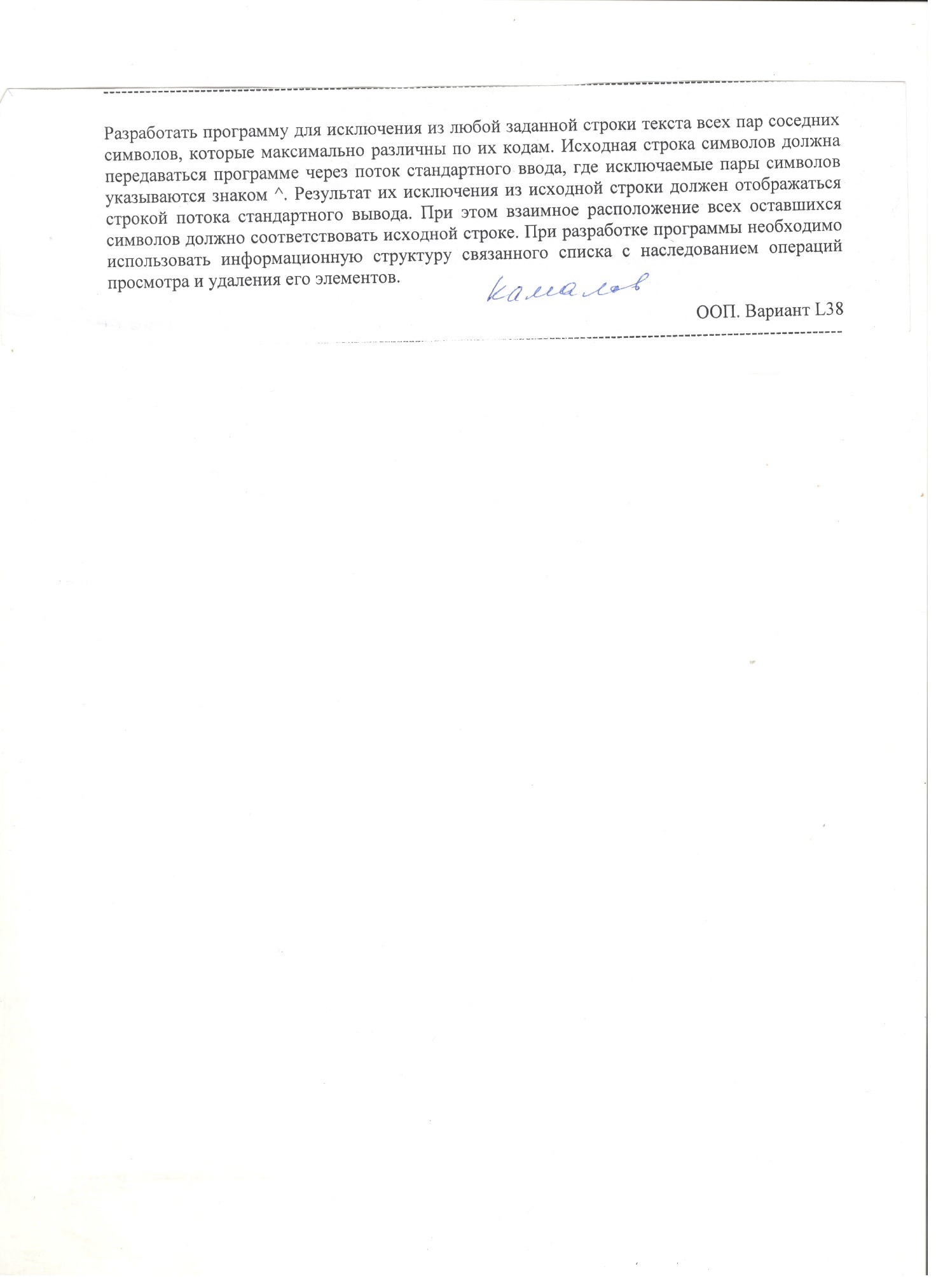
Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Козов А.В.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2023 г.*

**Задание**

****

**Описание программы**

**Входные данные:** символьная строка.

**Выходные данные:** изменённая символьная строка, где убраны пары с максимальной разницей в кодах.

**Класс Dlink**

*Информационные поля:*

\_next (Dlink\*) – указатель следующий элемент списка;

\_prev (Dlink\*) – указатель на предыдущий элемент списка;

*Конструкторы и деструкторы:*

Dlink () – конструктор по умолчанию;

*Методы:*

incr () – публичный метод, возвращающий указатель на следующий элемент списка;

toTail (int) – публичный метод получения указателя на элемент, двигаясь в направлении конца списка;

excluse () – публичный метод удаления текущего элемента из списка;

before (Dlink\*) – публичный метод вставки элемента перед текущим элементом списка.

**Класс SymLink**

*Информационные поля:*

\_let (char) – символ в списке;

*Конструкторы и деструкторы:*

SymLink (char) – конструктор с параметром;

*Методы:*

incr () – публичный перегруженный метод возвращающий указатель на следующий элемент списка, приведённый к типу SymLink\*;

print () – публичный метод печати символьной строки на экран и вычисления длины строки;

mx () – публичный метод поиска максимальной разницы кодов между соседними символами;

is\_max (int) – публичный метод проверки разницы между значениями символов текущего и следующего элементов списка на соответствие максимуму.

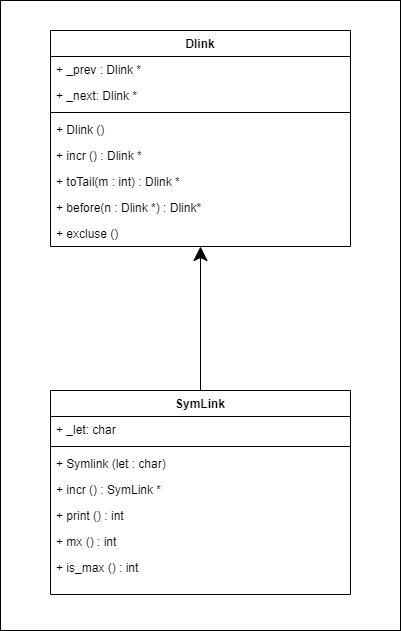


Рисунок 1. Диаграмма классов

**Алгоритм решения:**

1. Создаём объекты класса **SymLink** – head и tail, содержащие символы '\n' и связываем их между собой.
2. Считываем символы из стандартного потока ввода, создавая для каждого символа новый объект класса **SymLink** и помещая его перед объектом tail.
3. Вычисляем длину строки без учёта символов '\n'. Если она меньше 2, то программа завершает свою работу.
4. С помощью функции mx () вычисляем наибольшую разницу в кодах между соседними символами вызовом метода.
5. Инициализируем переменные count и i значениями 1 и устанавливаем указатель l на объект head.
6. Пока значение count меньше длины строки:

если разность кодов текущего символа и следующего символа равна max (для этого используется метод is\_max ()), указываем на эти две позиции символами '^', ставим указатель на текущий символ и удаляем его из списка с помощью метода exclude (). То же самое проделываем и со вторым символом. Иначе, переходим к следующему объекту **SymLink**, вызвав функцию incr ().

1. Освобождаем память, выделенную для объектов **SymLink**.

**Список литературы**

1. Пол Айра. Объектно-ориентированное программирование на C++: пер. с англ. 2-е изд. СПБ.: Невский Диалект; М.: Издательство БИНОМ, 2001. 462 с. [Pohl Ira. Object-Oriented Programming Using C++. 2nd ed. Addison-Wesley, 1996. 576 p.].
2. Страуструп Б. Язык программирования C++: пер. с англ / под ред. Н.Н. Мартынова. Специальное изд. М.: Бином, 2011. 1035 c. [Stroustrup B. The C++ Programming Language. Special ed. Addison-Wesley, 2000. 1029p.].
3. Шилдт Г. Самоучитель C++. 3-е изд. СПБ.: БХВ-Петербург, 2002. 688 с. [Schildt H. Teach Yourself C++. 3d ed. McGraw-Hill, 1998. 768 p.].
4. Волосатова Т.М., Родионов С.В. Объектно-ориентированное программирование на С++. Режим доступа: http://bigor.bmstu.ru/?cnt/?doc=VU/base.cou (дата обращения 24.04.2023).

**Приложение 1**

Текст программы

#include <iostream>

#include <cstdio>

#include <cstdlib>

class Dlink{

protected:

Dlink\* \_next;

Dlink\* \_prev;

public:

Dlink() : \_next(NULL), \_prev(NULL) {}

void excluse();

Dlink\* incr();

Dlink\* toTail(int);

Dlink\* before(Dlink\*);

};

class SymLink : public Dlink{

private:

char \_let;

public:

SymLink(char let) : Dlink(), \_let(let) {}

SymLink\* incr() {return (SymLink\*) Dlink::incr();}

int print();

int mx();

int is\_max(int);

};

int SymLink::mx(){

SymLink \*p = this->incr();

SymLink \*q;

int m = -1, s1;

while ((p->incr())->\_let != '\n'){

q = p->incr();

if (q != NULL && q->\_let != '\n'){

s1 = abs(p->\_let - q->\_let);

if (m < s1) m = s1;

}

p = q;

}

return m;

}

Dlink\* Dlink::toTail(int n){

Dlink\* p = this;

Dlink\* q = this;

int i = 0;

while(p != NULL){

q = p;

if (i == n) break;

p = p->\_next;

i++;

}

return q;

}

int SymLink::print(){

SymLink \*p = this;

SymLink \*q;

int n = 0;

while(p != NULL){

putchar(p->\_let);

q = p->incr();

p = q;

n++;

}

return(n);

}

Dlink\* Dlink::incr(){

return \_next;

}

void Dlink::excluse(){

if(\_next != NULL) \_next->\_prev = \_prev;

if(\_prev != NULL) \_prev->\_next = \_next;

return;

}

Dlink\* Dlink::before(Dlink\* p ){

p->\_next = this;

p->\_prev = \_prev;

if (\_prev != NULL) \_prev->\_next = p;

\_prev = p;

return p->\_prev;

}

int SymLink::is\_max(int max){

return (abs(\_let - this->incr()->\_let) == max);

}

int main(){

int max = 0;

int length = 0, ch, count = 1;

SymLink\* head;

SymLink\* tail;

SymLink\* q;

SymLink \*l;

head = new SymLink('\n');

tail = new SymLink('\n');

tail->before(head);

while((ch = getchar()) != '\n'){

q = new SymLink(ch);

tail->before(q);

}

if ((length = head->print() - 2) < 2){

std::cout << "No pairs!\n";

return 0;

}

max = head->mx();

int i = 1;

l = head;

while (count < length){

if (l->incr()->is\_max(max) == true){

printf("%\*c", i, '^');

printf("%c\n",'^');

for (int j = 0; j < 2; j++){

q = (SymLink\*) head->toTail(i);

q->excluse();

}

head->print();

count+=2;

} else {

l = l->incr();

count++;

i++;

}

}

q = head;

while (q != NULL){

l = q->incr();

q->excluse();

delete q;

q = l;

}

}

**Приложение 2**

Результат работы программы

