Условие:

Общая постановка. Пользовательский класс String должен содержать необходимые элементы-данные, которые создаются в динамической области памяти.

• Конструктор для создания строк: String (…);

• Деструктор: ~String();

• Метод ввода исходной строки: Set();

• Метод печати: void print(…);

Код методов – вне пространства определения класса. Программа иллюстрирует прямой и косвенный способы обращения к методам.

Ввести с клавиатуры строку символов S1. Признак окончания ввода строки - нажатие клавиши "Ввод". Программа должна содержать перегруженную операцию «=», использование которой скопирует S1 в S2.

Варианты заданий:

4.1. Длина L кратна 3-м, то удаляются все числа, делящиеся на 3;

4.2. Длина L делится на 5, то удаляется все символы кроме a-z

Исходный код программы:#define toDigit(c) (c-**'0'**)  
#define SIZE 255  
#define LINE cout << **"------------------------------------------------------\n"**#include **<iostream>**#include **<string>**#include **<stdlib.h>**#include **<istream>  
using namespace** std;  
  
**class** StringClass {  
 **char**\* str;  
 **char**\* str\_return;  
  
**public**:  
 StringClass();  
 StringClass(**char**\*);  
 ~StringClass();  
 **char**\* number1();  
 **char**\* number2();  
 **void** Set(**char**\*);  
 **friend void** print(StringClass&);  
 **friend** ostream& **operator**<<(ostream&, StringClass&);  
 **friend** istream& **operator**>>(istream&, StringClass&);  
 **friend char**\* number1(StringClass&);  
 **friend char**\* number2(StringClass&);  
};  
StringClass::StringClass()  
{  
 str = **new char**[SIZE];  
 str[0] = **'\0'**;  
 str\_return = **new char**[SIZE];  
 str\_return[0] = **'\0'**;  
  
};  
StringClass::StringClass(**char**\* s)  
{  
 str = **new char**[SIZE];  
 strcpy(str, s);  
 str\_return = **new char**[SIZE];  
 str\_return[0] = **'\0'**;  
};  
StringClass::~StringClass()  
{  
 **delete**[] str;  
 cout << **"\*Деструктор работает\*"** << endl;  
};  
**void** StringClass::Set(**char**\* s)  
{  
 **unsigned int** i;  
 **for** (i = 0; i < strlen(s); i++)  
 str[i] = s[i];  
 str[i] = **'\0'**;  
};  
  
*//Если длина L кратна 3-м, то удаляются все числа, делящиеся на 3***char**\* StringClass::number1()  
{  
 **bool** boolflag = **false**;  
 **int** j = 0;  
 **if** ((strlen(str) % 3) == 0){  
 cout<<**"--> Длина строки кратна 3\n"**;  
 **for** (**unsigned int** i = 0; i < strlen(str); i++){  
 **if** ((toDigit(str[i]) % 3) != 0){  
 boolflag = **true**;  
 str\_return[j] = str[i];  
 j++;  
 }  
 }  
 str\_return[j] = **'\0'**;  
 (boolflag == **true**) ? cout<<**"--> Есть цифры, кратные 3"**: cout<<**"--> Нет цифр, кратных 3"**;  
 }  
 **else** {  
 strcpy(str\_return, str);  
 cout<<**"--> Длина строки не кратна 3\n"**;  
 }  
 cout<<**"\n\*Преобразованная строка\*\n--> "**;  
  
 **return** str\_return;  
};  
  
*//Длина L делится на 5, то удаляется все символы кроме a-z***char**\* StringClass::number2()  
{  
 **int** i, j, f;  
 **char** b[SIZE]=**"0123456789"**;  
 **char** c[SIZE]=**".?:;|+-()&^%"**;  
   
 **if** ((strlen(str) % 5) ==0){  
 cout<<**"--> Длинна строки делится на 5\n"**;  
 **for**(**int** j=0;j<strlen(str);j++)  
 **for**(**int** i=0;i<strlen(str);i++){  
 **if**(str[j]==b[i]){  
 cout<<**"--> В строке присутствуют числа\n"**;  
 str[j]=**'#'**;  
 }  
 }  
 **for**( j=0;j<strlen(str);j++)  
 **for**(**int** f=0;f<strlen(str);f++){  
 **if**(str[j]==c[f]){  
 cout<<**"--> В строке присутствуют спец. символы\n"**;  
 str[j]=**'#'**;  
 }  
 }  
 }  
 **else** cout<<**"--> Длинна строки не делится на 5\n"**;  
 **return** str;  
};  
  
**char**\* number1(StringClass& obj) {  
 **return** obj.number1();  
};  
  
**char**\* number2(StringClass& obj) {  
 **return** obj.number2();  
};  
  
**void** print(StringClass& obj) {  
 cout << obj.str << **" "** << obj.str\_return;  
};  
  
ostream& **operator**<<(ostream& stream, StringClass& ob)  
{  
 stream << ob.str;  
 **return** stream;  
};  
istream& **operator**>>(istream& stream, StringClass& ob)  
{  
 stream >> ob.str;  
 **return** stream;  
};  
**int** main()  
{  
 **char** s[265];  
 **char** s1[265];  
  
 LINE;  
 cout<<**"Задание №1\n"**;  
 cout<<**"|Пример с доступом к методам класса непосредственно через переменную|\nВведите строку и нажмите \"Enter\" => "**;  
 cin.getline(s, 256);  
 StringClass str(s);  
 cout<<**"Вы ввели строку: "**;  
 print(str);  
 cout<<**"\n\n\*Результат выполнения\*"**<<endl;  
 cout<<number1(str)<<endl;  
  
 LINE;  
 cout<<**"Задание №1\n"**;  
 cout << **"|Пример с доступом к методам класса через указатель|\nВведите строку и нажмите \"Enter\" => "**;  
 cin.getline(s,256);  
 StringClass \*pstr;  
 pstr=**new** StringClass();  
 pstr->Set(s);  
 cout<<**"Вы ввели строку: "**;  
 print(\*pstr);  
 cout<<**"\n\n\*Результат выполнения\*"**<<endl;  
 cout<<number1(\*pstr)<<endl;  
 **delete** pstr;  
  
 LINE;  
 cout<<**"Задание №2\n"**;  
 cout<<**"|Пример с доступом к методам класса непосредственно через переменную|\nВведите строку и нажмите \"Enter\" => "**;  
 cin.getline(s1, 256);  
 StringClass str1(s1);  
 cout<<**"Вы ввели строку: "**;  
 print(str1);  
 cout<<**"\n\n\*Результат выполнения\*"**<<endl;  
 cout<<number2(str1)<<endl;  
  
 LINE;  
 cout<<**"Задание №2\n"**;  
 cout << **"|Пример с доступом к методам класса через указатель|\nВведите строку и нажмите \"Enter\" => "**;  
 cin.getline(s,256);  
 StringClass \*pstr1;  
 pstr1=**new** StringClass();  
 pstr1->Set(s);  
 cout<<**"Вы ввели строку: "**;  
 print(\*pstr1);  
 cout<<**"\n\n\*Результат выполнения\*"**<<endl;  
 cout<<number2(\*pstr1)<<endl;  
 **delete** pstr1;  
 **return** 0;  
};

Скриншоты программы:

