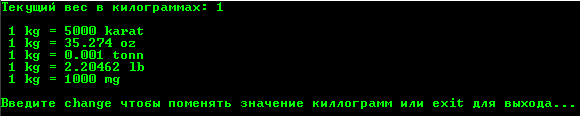
**Содержание:**

**Программа №1**

* 1. **Постановка задачи**
* Разработать класс **Конвертер весов**.
* Класс должен хранить вес в килограммах и предоставлять методы по его преобразованию в другие единицы измерения (например, аптечный фунт, тройская унция, пуд, …).
* Класс должен предоставлять операции: 1) установить текущий вес в килограммах, 2) узнать текущий вес в килограммах, 3) узнать текущий вес в выбранной единице измерения (из списка поддерживаемых).
* Класс должен содержать все необходимые конструкторы, оператор присваивания, а также «уметь» выводить себя на консоль.

**1.2 руководство пользователя**

**1.2.1** Начальный экран

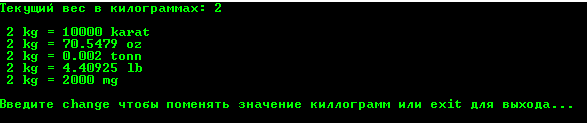
****

**1.2.2** Введите “change” или “exit”

**1.2.3** Если ты ввел “change”, то ты молодец, и можешь поменять значение килограмм



**1.2.4** Наслаждайся полученной информацией



* 1. **Руководство программиста**

Класс **KON** содержит методы , которые реализуют перевод величин веса из килограмм в караты, миллиграммы, тонны, фунты, унции и всё.

Методы возвращающие значение в соответствующих величинах:

**double Karat()**

**double mg()**

**double Tonn()**

**double lb()**

**double oz()**

**double GetKG()**

Метод изменяющий значение :

**double SetKG(double kg)**

Конструктор универсальный:

KON (double kg = 1.0)

{

this -> kg = kg;

}

**1.4 Эксперимент**

В результате эксперимента, показанного на рис.1 и рис. 2, мы видим, что в итоге выводится вес преобразованный из килограммов в другие единицы измерения. Программа функционирует и работает правильно.



Рис. 1

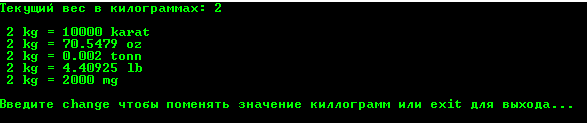


Рис. 2

**1.5 Список литературы**

* http://cppstudio.com/post/6964/
* https://code-live.ru/post/cpp-classes/
* http://codelessons.ru/cplusplus/klassy-v-c-rukovodstvo-dlya-nachinayushhix.html

**Программа №2**

**2.1 Постановка задачи**

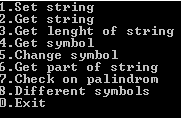
* Разработать класс **Строка**.
* Класс должен хранить строку символов произвольной длины (от 0 до 40).
* Класс должен предоставлять следующие операции: 1) задать строку, 2) узнать длину строки, 3) получить символ строки по его индексу, 4) изменить символ строки по заданному индексу, 5) выделить подстроку из строки, 6) проверить, является ли строка палиндромом, 7) найти, сколько разных символов латинского алфавита содержится в строке.
* Класс должен содержать все необходимые конструкторы, деструктор, оператор присваивания, а также «уметь» выводить себя на консоль.

**2.2 руководство пользователя**

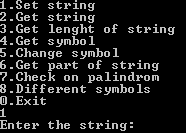
**2.2.1** При запуске введите вашу строчку



**2.2.2** Выбирайте пункт меню



**2.2.2.1** Изменение строки



**2.2.2.2** Получение строки

**2.2.2.3** Получение количества элемента

**2.2.2.4** Получение определенного символа

**2.2.2.5** Изменение символа

**2.2.2.6** Получение подстроки

**2.2.2.7** Проверка на палиндром

**2.2.2.8** Определение сколько разных символов латинского алфавита содержится в строке

**2.2.2.0** Выход из программы

**2.3 руководство программиста**

Class Line

int length – длина строки

char \*string – указатель на строку

Содержит 2 конструктора:

Line() – выделяет память под строку

Line(const Line &obj2) – конструктор копирования

Деструктор:

~Line() – освобождает память

И такие метода как:

**void SetString(char\* word)**

**char\* GetString()**

**int GetStringLength()**

**bool isPalindrom()**

**char GetSymbol(int index)**

**void ChangeSymbolFromString(int index2, char change)**

**char GetPartOfString(int k)**

**int HowManyDifferentLatinSymbolsInString()**

Так же есть оператор присваивания

**Line& operator=(Line &obj1)**

**Список литературы**

* http://cppstudio.com/post/6964/
* https://code-live.ru/post/cpp-classes/
* http://codelessons.ru/cplusplus/klassy-v-c-rukovodstvo-dlya-nachinayushhix.html
* http://cppstudio.com/post/432/
* http://codelessons.ru/cplusplus/dinamicheskie-massivy-i-peremennye-vse-samoe-glavnoe.html

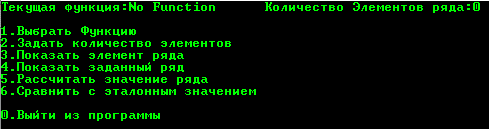
**Программа №3**

**3.1 Постановка задачи**

* Разработать класс **Ряд Тейлора**.
* Класс должен формировать ряд Тейлора для выбранной функции из списка поддерживаемых с заданным числом членов ряда. Минимальный список функций: sin(x), cos(x), exp(x).
* Класс должен содержать необходимые служебные методы (конструкторы, деструктор и пр.).
* Класс должен предоставлять следующие операции: 1) задать текущую функцию, 2) узнать текущую функцию, 3) задать текущее число членов ряда, 4) узнать текущее число членов ряда, 5) выдать формулу ряда для выбранной функции, 6) выдать значение заданного члена ряда, 7) рассчитать значение ряда в выбранной точке x, 8) вывести отклонение значения ряда в выбранной точке от эталонного значения текущей функции в данной точке (эталонное значение вычисляется, используя соответствующую функцию из стандартной библиотеки C++).

**3.2 руководство пользователя**

**3.2.1** При открытии программы вы попадаете в главное меню

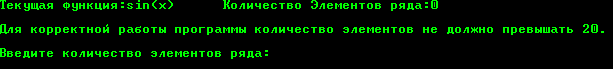


**3.2.2** Выбирайте пункты меню

**3.2.2.1** Задайте функцию из списка



**3.2.2.2** Задайте количество элементов



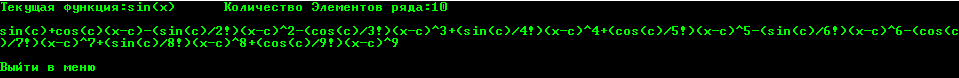
**3.2.2.3.1** Введите номер требуемого элемента ряда Тейлора



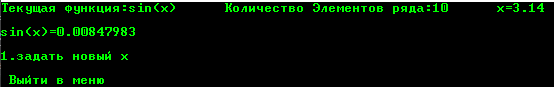
**3.2.2.3.2** Получите результат



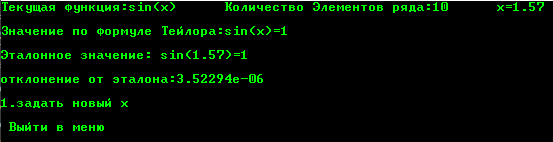
**3.2.2.4** Получение всего ряда Тейлора



**3.2.2.5** Получить результат ряда Мак Лорена



**3.2.2.6** Сравнить с эталонным значением



**3.2.2.0** Выйти из программы

**3.3 руководство программиста**

Класс **Taylor** хранит в себе такие данные как:

**int Function;//текущая функция(1-sin(x) 2-cos(x) 3-exp(x))**

**int number=0;//количество членов ряда**

**double x=0.0;//точка х в формуле Маклорена**

**double const c = 0.0;//точка для формулы Тейлора**

**//bool vr;//вывод(0) или расчет функции(1)**

**long double ans;//результат подсчета**

Cодержит в себе три основных метода:

**void SinTaylor(bool vr = 0, int number = 0, int a = -1)**

**void CosTaylor(bool vr = 0, int number = 0, int a = -1)**

**void ExpTaylor(bool vr = 0, int number = 0, int a = -1)**

которые выполняют основную работу класса – выводят и считают ряд Тейлора.

Данные в классе задаются с помощью методов:

**void SetFunction(int Function)**

**void SetNumber(int number)**

**void SetPoint(double x)**

Данные в классе выводятся с помощью методов:

**long double GetAns()**

**void GetFunction()**

**int GetFunctionint()**

**int GetNumber()**

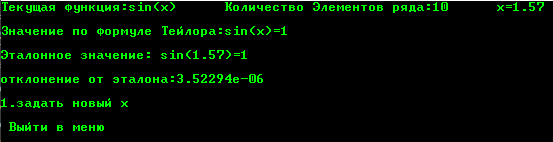
**void GetElemen ( int number=0/\*номер элемента\*/)**

**void GetSequence()**

**double GetPoint()**

**3.4 Эксперимент**

В результате эксперимента, показанного на последующих рисунках, мы видим, что словарь и методы, требуемые в постановке задачи, работают исправно и без тормозов. Программа функционирует и работает правильно.



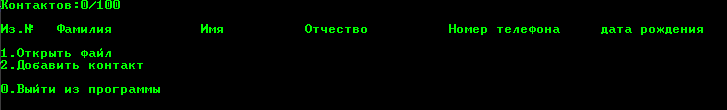
**Программа №4**

**4.1 Постановка задачи**

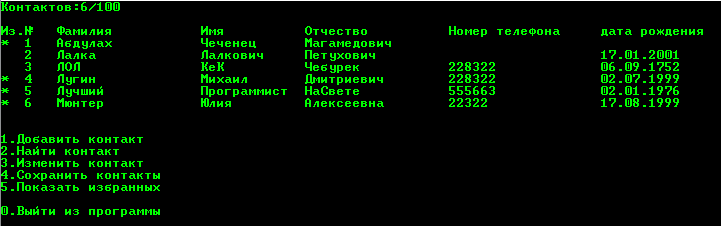
* Разработать класс **Контакты**.
* Класс должен хранить информацию о контактах владельца. Каждый контакт содержит следующие данные: фамилия; имя; отчество; телефон; день рождения (день, месяц, год); признак, относится ли контакт к избранным. Контакты хранятся упорядоченно по фамилии, имени, отчеству. Фамилия, имя, отчество (ФИО) являются обязательными полями. Данные вводятся на русском языке.
* Класс должен содержать необходимые служебные методы.
* Класс должен предоставлять следующие операции: 1) создать новый контакт, 2) изменить выбранный контакт, 3) найти контакт по ФИО, 4) найти контакт по телефону, 5) выдать все контакты на заданную букву, 6) узнать текущее число контактов, 7) внести контакт в список избранных, 8) удалить контакт из списка избранных, 9) выдать все избранные контакты, 10) удалить контакт, 11) сохранить контакты в файл, 12) считать контакты из файла.

**4.2 Руководство пользователя**

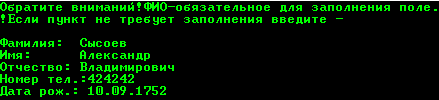
**4.2.1** После открытии программы выбирайте добавить новый контакт или открыть файл



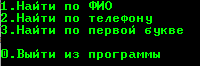
**4.2.2** Далеебудет доступны следующие пункты

****

**4.2.2.1** Добавление контакта происходит пошагово



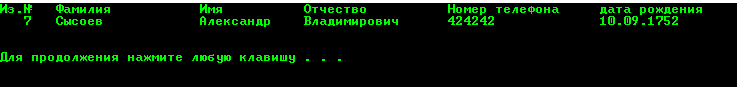
**4.2.2.2** Найти контакты можно тремя способами



В поиске по ФИО вводим любую часть имени



Если такие контакты нашлись то они выводятся на экран

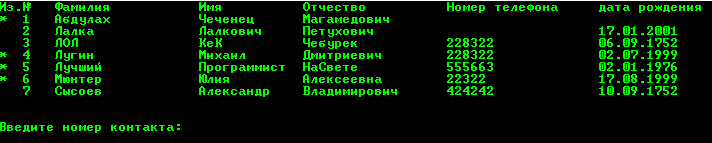


Если не нашлись, то ничего не выведется

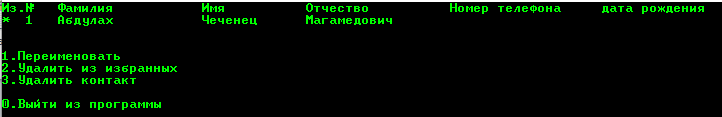




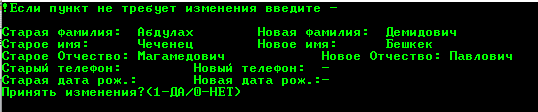
**4.2.3.1** Для изменения контакта введите его уникальный номер

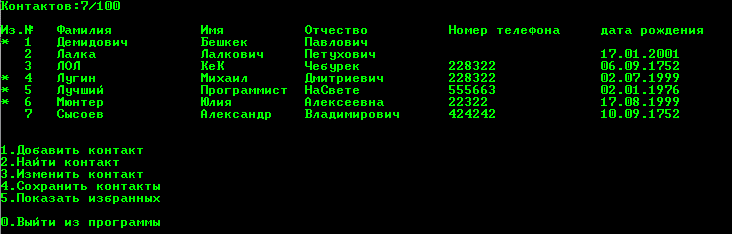
****

**4.2.3.2** Далее откроется меню изменений



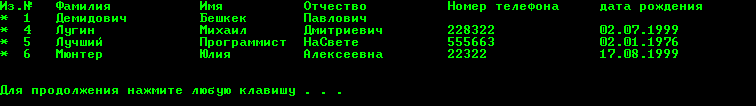
Выбираете и контакт изменен





**4.2.4** При сохранении контактов, они появятся в следующий раз при открытии программы !Внимание! если контакты не были сохранены то они будут утеряны после закрытия программы.

**4.2.5** Показывает избранные контакты



**4.2.0** Выход из программы

**Руководство программиста**

Класс **Contact** хранит в себе такие данные :

**int count=0;**

**string Surname[Max];**

**string Name[Max];**

**string SecondName[Max];**

**string PhoneNumber[Max];**

**string BD[Max];//день рождения**

**bool FAV[Max];//избранный**

**string ALFAVIT = RUString();**

**string FILE = "Contacts.txt";**

Данные задаются с помощью методов:

**void SetSurname(string Surname, int i)**

**void SetName(string Name, int i)**

**void SetSecondName(string SecondName, int i)**

**void SetPhoneNumber(string PhoneNumber, int i)**

**void SetBD(string BD, int i)**

**void SetFAV(bool FAV, int i)**

Данные выводятся с помощью методов:

**void ShowContacts()**

**void ShowContact(int i)**

**void ShowFAV()**

**int GetCount()**

**string GetSurname(int i)**

**string GetName(int i)**

**string GetSecondName(int i)**

**string GetPhoneNumber(int i)**

**string GetBD(int i)**

**bool GetFAV( int i)**

**void UpOut()**

Вся логика находится в методах:

**void AddContact(string Surname,string Name,string SecondName,string PhoneNumber,string BD)**

**void DeleteContact(int i)**

**void FAVplus(int i)**

**void FAVminus(int i)**

**void SetContact(int i ,string a[5])**

**void Sort()**

**void Search(string a)**

**void SearchChar(char a)**

**void SearchPhone(string a)**

Данные методы сохраняют и выводят данные :

**void FILEOpen()**

**void FILESave()**

**Список литературы**

* http://cppstudio.com/post/6964/
* https://code-live.ru/post/cpp-classes/
* http://codelessons.ru/cplusplus/klassy-v-c-rukovodstvo-dlya-nachinayushhix.html
* http://cppstudio.com/post/432/
* http://codelessons.ru/cplusplus/dinamicheskie-massivy-i-peremennye-vse-samoe-glavnoe.html
* http://cppstudio.com/post/446/
* http://purecodecpp.com/archives/2751
* <http://cppstudio.com/post/9033/>

**Программа №5**