

## Задача (Task) 3

### 1. Расчет интегралов

Разработать класс **Integrals**.

Класс должен позволять вычислять приближенное значение интеграла от произвольной функции одной переменной, заданной в виде функции языка C++.

Класс должен для заданного интервала интегрирования  $[a, b]$  и количества отрезков разбиения интервала  $n$  вычислять приближенное значение интеграла, используя методы левых, правых и средних прямоугольников.

Класс должен содержать необходимые конструкторы и деструктор.

Класс должен предоставлять следующие операции (методы класса):

- 1) конструктор инициализатор с параметрами: интервал интегрирования, число отрезков разбиения интервала интегрирования;
- 2) задать интегрируемую функцию;
- 3) вычислить значение интеграла указанным методом;
- 4) вывести результат вычисления на экран.

### 2. Табулятор функции

Разработать класс **Tabulator**.

Класс должен позволять выполнять табулирование произвольной функции одной переменной, заданной в виде функции  $f(x)$  языка C++.

Табулирование функции  $f(x)$  на заданном интервале табулирования  $[a, b]$  и заданном количестве точек табуляции  $n$  состоит в формировании динамических массивов значений  $X$  в равноотстоящих точках табуляции и значений функции  $F$  в этих точках в виде:

$X: x_0=a, \quad x_1=a+h, \quad x_2=a+2h, \quad \dots \quad x_{n-1}=b$   
 $F: f_0=f(x_0), \quad f_1=f(x_1), \quad f_2=f(x_2), \quad \dots \quad f_{n-1}=f(x_{n-1}),$

Где  $h = (b - a) / (n - 1)$

Для вычисления значений табулированной функции использовать в качестве тестовой одну из элементарных функций:  $\sin, \cos, \exp$

Вычисление табулированной функции в точке  $x$  ( $a \leq x \leq b$ ) выполняется с помощью линейной интерполяции по двум точкам плоскости:  $(x_i, f_i)$  и  $(x_{i+1}, f_{i+1})$ , где:  $x_i \leq x \leq x_{i+1}$ .

Класс должен содержать необходимые конструкторы и деструктор.

Класс должен предоставлять следующие операции (методы класса):

- 1) выполнить табулирование для заданной тестовой функции,
- 2) вычислить значение табулированной функции в заданной точке  $x$  и сравнить это значение со значением тестовой функции в этой точке.
- 3) перегрузить операцию записи – чтения из потока (файлового)

### 3. Ряд Тейлора

Разработать класс **Taylor**.

Ряд Тейлора – приближенное представление значения функции  $f(x)$  в окрестности точки  $a$  в виде суммы в виде конечной суммы членов ряда Тейлора:

$$f(x) = f(a) + \frac{f'(a)}{1!}(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x - a)^n + \dots$$

Класс должен формировать динамический массив значений коэффициентов ряда (типа `double`) Тейлора:  $f(a), f'(a)/1!, f''(a)/2!, \dots$  для заданного числа членов ряда. В качестве функции на усмотрение студента может быть выбрана одна из:  $\sin(x), \cos(x), \exp(x)$ .

Класс должен содержать необходимые конструкторы и деструктор.

Класс должен предоставлять следующие операции (методы класса):

- 1) задать значение параметра  $a$  и подсчитать значения массива коэффициентов
- 2) выдать формулу ряда с коэффициентами, подсчитанными для заданного параметра  $a$ ,
- 3) выдать значение заданного члена ряда,
- 4) рассчитать значение ряда в выбранной точке  $x$ ,
- 5) вывести отклонение значения ряда в выбранной точке от эталонного значения функции в данной точке (эталонное значение вычисляется, используя соответствующую функцию из стандартной библиотеки C++).
- 6) перегрузить операцию записи – чтения из потока (файлового)

### 4. Пользовательское меню

Разработать класс **Menu**

Класс должен предоставлять одноуровневое (однострочное) меню с заданным числом команд в консольном режиме экрана.

Класс должен содержать необходимые конструкторы, деструктор и методы (по выбору студента).

В качестве обязательных, класс должен содержать методы:

- 1) Конструктор инициализатор с параметрами:
  - a. Местоположение меню: координаты начала строки меню и длину меню.
  - b. Количество пунктов меню
  - c. Массив названий пунктов меню
- 2) Конструктор с параметром – имя текстового файла описания меню. Текст описания меню должен содержать:
  - a. Местоположение меню в виде: xCor yCor size
  - b. Количество пунктов меню
  - c. Названия пунктов (по одному в строке)

**Пример:**

1 1 80

3

File

Main

View

В этом примере указано, что меню:

- начинается в позиции экрана с координатами (1, 1) и имеет длину 80 символов;

- включает 3 пункта;

- которые имеют имена: File, Main и View.

- 3) Вывести (показать) меню на экран,
- 4) Удалить меню с экрана,
- 5) Выбрать пункт меню с помощью клавиш: стрелки (влево, вправо) и ввод (Enter) с выдачей номера выбранного пункта меню.

Работа с консолью - См. <http://www.c-cpp.ru/funkcii/conioh>

## 5. Однострочный текстовый редактор

Разработать класс **TextEditor**.

Класс должен предоставлять возможность разместить в выбранной позиции окна консоли поле заданной длины для ввода с клавиатуры последовательности символов. В минимальном варианте длина последовательности не должна превышать длину поля ввода.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор и пр.).

Класс должен предоставлять следующие операции:

- 1) конструктор инициализатор с параметрами: начальная позиция поля ввода (x, y) в окне консоли. длина поля ввода;
- 2) показать текстовый редактор, убрать с экрана текстовый редактор;
- 3) обеспечить ввод пользователем строки с длиной не больше длины поля ввода;
- 4) выдать введенную пользователем строку.

Программа должна иметь простейшее меню:

- Введите положение и размер окна

- Показать редактор

- Убрать редактор

- Введите и отредактируйте текст

- Завершить работу

Работа с консолью - См. <http://www.c-cpp.ru/funkcii/conioh>

## 6. Словарь переводчика

Разработать класс **Dictionary**.

Класс должен предоставлять возможность формировать англо-русский словарь. В этом словаре каждому английскому слову соответствует ровно одно русское слово-перевод.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор и пр.).

Класс должен предоставлять следующие операции:

- 1) добавить в словарь слово и его перевод,
- 2) изменить перевод указанного слова,
- 3) узнать перевод выбранного слова,
- 4) проверить наличие слова в словаре,
- 5) узнать число слов в словаре,
- 6) перегрузить операции записи, чтения словаря в поток.
- 7) перегрузить операции присваивания и объединения словарей.