ТРЕБОВАНИЯ
К ПРОГРАММАМ “ПРАКТИКУМА НА ЭВМ”
Первое задание по приближениям функций 1-й переменной

1. Программа должна получать **все начальные параметры в качестве аргументов командной строки**. Программа имеет 4 обязательных аргумента:

   (1) a – левый конец отрезка (тип double),
   (2) b – правый конец отрезка (тип double),
   (3) n – начальное значение для числа точек интерполяции (тип int),
   (4) k – начальное значение номера приближаемой функции (тип int).

2. В программе должны быть реализованы подпрограммы для задания следующих приближаемых функций $f(x)$ по аналитически заданной формуле в зависимости от параметра k:

   (1) для $k = 0$ $f(x) = 1$
   (2) для $k = 1$ $f(x) = x$
   (3) для $k = 2$ $f(x) = x^2$
   (4) для $k = 3$ $f(x) = x^3$
   (5) для $k = 4$ $f(x) = x^4$
   (6) для $k = 5$ $f(x) = e^x$
   (7) для $k = 6$ $f(x) = 1/(25x^2 + 1)$

   Если в задаче требуются производные функции f, то они **вычисляются по правилам дифференцирования** и реализуются в виде подпрограммы.

3. В программе должны быть реализованы два метода (1), (2), выданные преподавателем. **Оба метода** должны быть реализованы в одной программе и их результаты должны показываться в графическом интерфейсе программы. Обычно метод 1 – это метод многочленной аппроксимации, метод 2 – метод кусочно-многочленной аппроксимации.

4. Построение приближающей функции (многочлена или кусочно-многочленной функции) должно быть оформлено в виде подпрограммы, находящейся в отдельном файле и получающей в качестве аргументов

   (a) количество точек приближения n (в которых известно значение функции);
   (b) массив x длины n с точками приближения $x_1, \ldots, x_n$;
   (c) массив f длины n со значениями $f_1, \ldots, f_n$ функции f в точках приближения;
   (d) массив a требуемой длины для сохранения результатов работы этой функции – коэффициентов приближающей функции;
   (e) дополнительные вектора, если алгоритму требуется дополнительная память.

   Получать в этой подпрограмме дополнительную информацию извне через глобальные переменные, включаемые файлы и т.п. запрещается.

5. Вычисление значения приближающей функции (многочлена или кусочно-многочленной функции) в точке должно быть оформлено в виде подпрограммы, находящейся в том же файле, что и предыдущая функция, и получающей в качестве аргументов

   (a) точку x, в которой требуется вычислить значение приближающей функции;

(b) отрезок $[a, b]$, на котором построено значение приближающей функции;
(c) количество точек приближения n;
(d) массив x длины n с точками приближения $x_1, \ldots, x_n$;
(e) массив a с коэффициентами приближающей функции.

Получать в этой подпрограмме дополнительную информацию извне через глобальные переменные, включаемые файлы и т.п. запрещается.

6. Программа должна содержать подпрограмму графического представления заданной функции в окне приложения, разработанного с помощью **библиотеки Qt5**. Эта подпрограмма должна находиться в отдельном файле и получать в качестве аргументов

   (a) отрезок $[a, b]$, на котором требуется построить график функции;
   (b) указатель на функцию, вычисляющую значение рисуемой функции в точке.

   Функция должна:

   (a) вычислять максимальное значение функции на области рисования и **осуществлять масштабирование (независимое по X и Y)** для того, чтобы график не выходил за границы окна и не оказался слишком мелким);
   (b) выводить **на графический экран и в консоль** максимальное по модулю значение функции.

7. Интерфейсная часть программы по нажатию указанной клавиши **должна**:

   (a) По нажатию клавиши 

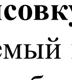
 циклически **менять номер** k **приближаемой функции** и **перерисовывать новый график**. Значение номера приближаемой функции k, а также текстовое представление функции **должно выводиться** в графическом окне (например, выводится `k=4 f(x)=x^4`).

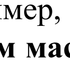
   (b) По нажатию клавиши 

 циклически **менять состав отображаемых графиков** и **перерисовывать новый график**:

      i. показывать график функции и график ее приближения по методу (1);
      ii. показывать график функции и график ее приближения по методу (2);
      iii. показывать график функции и графики ее приближения по методам (1) и (2);
      iv. показывать графики погрешностей методов (1) и (2);

   Каждый наборов графиков отображается **в своем масштабе, причем разном для осей X и Y**, так, чтобы отрезок $[a, b]$ был равен **ширине области рисования окна**, а отрезок $[F_{min}, F_{max}]$ был равен **высоте области рисования окна**; здесь F_{min} – минимальное значение визуализируемых графиков на отрезке $[a, b]$, F_{max} – максимальное значение визуализируемых графиков на отрезке $[a, b]$. Значение величины $\max\{|F_{min}|, |F_{max}|\}$ **должно выводиться** как в графическом окне, так и на текстовой консоли.

   (c) По нажатию клавиши 

 увеличивать, а по нажатию клавиши 

 уменьшать масштаб текущего графика, осуществляя **двукратное растяжение/сжатие** оси X относительно середины отрезка $[a, b]$ и **перерисовку графика в новом масштабе**. Например, если s раз нажать клавишу 2, то визуализируемый набор графиков отображается **в своем масштабе, причем разном для осей X и Y**, так, чтобы отрезок $[a/2^s, b/2^s]$ был равен **ширине области рисования окна**, а отрезок $[F^{(s)}_{min}, F^{(s)}_{max}]$ был равен **высоте области рисования окна**; здесь $F^{(s)}_{min}$ – минимальное значение визуализируемых графиков на отрезке $[a/2^s, b/2^s]$, $F^{(s)}_{max}$ – максимальное значение визуализируемых графиков на отрезке $[a/2^s, b/2^s]$. Значение величины $\max\{|F_{min}^{(s)}|, |F_{max}^{(s)}|\}$, а также значение величины текущего масштаба s **должно выводиться** в графическом окне.

(d) По нажатию клавиши 

увеличивать, а по нажатию клавиши 

уменьшать в 2 раза **число точек приближения** n и **перерисовывать графики для нового числа точек приближения**. Значение текущего числа точек n **должно выводиться** в графическом окне. Для методов многочленной аппроксимации при $n > 50$ приближение не строится и не отображается.

(e) По нажатию клавиши 

прибавлять, а по нажатию клавиши 

вычитать к/от вычисленному значению функции $f_{n/2} = f(x_{n/2})$ одну десятую максимума функции f на отрезке $[a, b]$, моделируя погрешность измерения, и **перерисовывать новый график**. Например, если p раз нажать клавишу 6, то все приближения и графики строятся на для функции $f(x)$, а для функции $\hat{f}(x)$, где

$$\hat{f}(x) = \begin{cases} f(x) & x \neq x_{n/2} \\ f(x) + p * 0.1 * \max\limits_{[a,b]} |f| & x = x_{n/2} \end{cases}$$

Значение текущего возмущения p **должно выводиться** в графическом окне.

8. Реализованные в программе методы интерполяции должны проходить, как минимум, следующие проверки

   (a) **Быть точными на многочлене "правильной" степени**. Быть точным означает, что для минимально возможного n (например, $n = 5$) погрешность метода на таком многочлене имеет порядок машинной точности. Все методы, рассматриваемые в курсе, точны на многочленах степени 0 и 1. Для каждого метода из описания вытекает степень многочлена, на которой он точен.

   (b) **Погрешность метода должна падать в "правильное" число раз при удвоении** n. Асимптотическое поведение точности метода указано в его описании. Асимптотику надо проверять для достаточно больших n, обычно 50–100.

   (c) **Методы кусочно многочленной аппроксимации должны практически мгновенно работать для** $n = 10^7$. Как время работы метода, так и время обновления экрана не должны превышать 1 секунды даже на компьютерах десятилетней давности для $n = 10^7$ (десять миллионов точек интерполяции).

   (d) **Скорость перерисовки экрана при изменении размера окна для** $n = 10^7$ **должна быть практически мгновенной**. Время обновления окна не должно превышать 1 секунды даже на компьютерах десятилетней давности без аппаратного графического ускорителя для $n = 10^7$ (десять миллионов точек интерполяции).

   (e) **Не должно быть утечек памяти в самой программе**. Утечки в используемой библиотеке Qt5 допустимы.

   (f) **Программа должна быть самостоятельно написанной, как метод, так и графический интерфейс**. Не должно быть сходства с вариантами из сети Интернет.