ТРЕБОВАНИЯ
К ПРОГРАММАМ “ПРАКТИКУМА НА ЭВМ”
Первое задание по приближениям функций 2-х переменных

1. Программа должна получать **все начальные параметры в качестве аргументов командной строки**. Программа имеет 6 обязательных аргументов:

   (1) "имя файла" – имя файла, содержащего спецификацию области (строка),
   (2) nx – начальное значение для числа точек интерполяции по оси X (тип int),
   (3) ny – начальное значение для числа точек интерполяции по оси Y (тип int),
   (4) k – начальное значение номера приближаемой функции (тип int),
   (5) ε – точность решения системы линейных уравнений (тип double),
   (6) p – число вычислительных потоков (тип int).

2. Формат файла, содержащего спецификацию области

   - Строки, начинающиеся с символа `#`, игнорируются (служат для задания комментариев)
   - Пустые строки (содержащие только пробельные символы и завершающий символ `\n`) игнорируются
   - Параметрами являются числа (с плавающей точкой), разделенные пробельными символами или `\n`
   - Пробельные символы: пробел, табуляция

   Концом спецификации считается конец файла со спецификацией. В качестве тестовой области рекомендуется брать область, лежащую в круге $x^2 + y^2 \leq 1$, чтобы избежать большой амплитуды приближаемой функции. Пример файла:

```
# Задание области - прямоугольника
# Верхний левый угол, (x,y)
-0.5 0.5
# Нижний правый угол
0.5 -0.5
```

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

   (1) для $k = 0$ $f(x, y) = 1$
   (2) для $k = 1$ $f(x, y) = x$
   (3) для $k = 2$ $f(x, y) = y$
   (4) для $k = 3$ $f(x, y) = x + y$
   (5) для $k = 4$ $f(x) = \sqrt{x^2 + y^2}$
   (6) для $k = 5$ $f(x) = x^2 + y^2$
   (7) для $k = 6$ $f(x) = e^{x^2 - y^2}$
   (8) для $k = 7$ $f(x) = 1/(25(x^2 + y^2) + 1)$

4. Построение приближающей функции (многочлена или кусочно-многочленной функции) должно быть оформлено в виде подпрограммы, находящейся в отдельном файле. 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 отдельном файле. 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. Функция должна:

   (a) вычислять максимальное значение функции на области рисования и **осуществлять масштабирование (независимое по XY и Z)** 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 0 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=3 f(x,y)=x+y`).

   (b) По нажатию клавиши 1 циклически **менять состав отображаемых графиков** и **перерисовывать новый график**:

      i. показывать график функции;
      ii. показывать график ее приближения;
      iii. показывать график погрешности приближения.

   Каждый из графиков отображается **в своем масштабе, причем разном для осей XY и Z**, так, чтобы поверхность, образованная графиком функции над заданной областью, была вписана в окно рисования. Значение величины $\max\{|F_{min}|, |F_{max}|\}$ **должно выводиться** как в графическом окне, так и на текстовой консоли, где F_{min} – минимальное значение визуализируемого графика в области, F_{max} – максимальное значение визуализируемого графика в области.

   (c) По нажатию клавиши 2 увеличивать, а по нажатию клавиши 3 уменьшать масштаб текущего графика, осуществляя **двукратное растяжение/сжатие** осей XY относительно центра тяжести области и **перерисовку графика в новом масштабе**. Например, если s раз нажать клавишу 2, то визуализируемый график отображается **в своем масштабе, причем разном для осей XY и Z**, так, чтобы поверхность, образованная графиком функции над $1/2^s$ частью заданной области, была вписана в окно рисования. Значение величины $\max\{|F^{(s)}_{min}|, |F^{(s)}_{max}|\}$, а также значение величины текущего масштаба s **должно выводиться** в графическом окне, где $F^{(s)}_{min}$ – минимальное значение визуализируемого графика в $1/2^s$ части области, $F^{(s)}_{max}$ – максимальное значение визуализируемого графика в $1/2^s$ части области.

   (d) По нажатию клавиши 4 увеличивать, а по нажатию клавиши 5 уменьшать в 2 раза **число точек приближения** nx, ny и **перерисовывать графики для нового числа точек приближения**. Значение текущего числа точек nx, ny **должно выводиться** в графическом окне.

   (e) По нажатию клавиши 6 прибавлять, а по нажатию клавиши 7 вычитать к/от вычисленному значению функции $f_{n/2} = f(x_{nx/2}, y_{ny/2})$ одну десятую максимума функции f в исходной области, моделируя погрешность измерения, и **перерисовывать новый график**. Например, если p раз нажать клавишу 6, то все приближения и графики строятся не для функции $f(x, y)$, а для функции $\hat{f}(x, y)$, где

$$\hat{f}(x,y) = \begin{cases} f(x,y) & (x,y) \neq (x_{nx/2}, y_{ny/2}) \\ f(x,y) + p * 0.1 * \max\limits_{\Omega} |f| & (x,y) = (x_{nx/2}, y_{ny/2}) \end{cases}$$

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

(f) По нажатию клавиши  вращать по часовой стрелке, а по нажатию клавиши  вращать против часовой стрелки вокруг оси OZ на угол $\pi/12$. Значение текущего угла поворота **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 nx, ny (например, $nx = ny = 5$) 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 nx и ny**. Асимптотическое поведение точности метода указано в его описании. Асимптотику надо проверять для достаточно больших nx, ny, обычно 50–100.

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