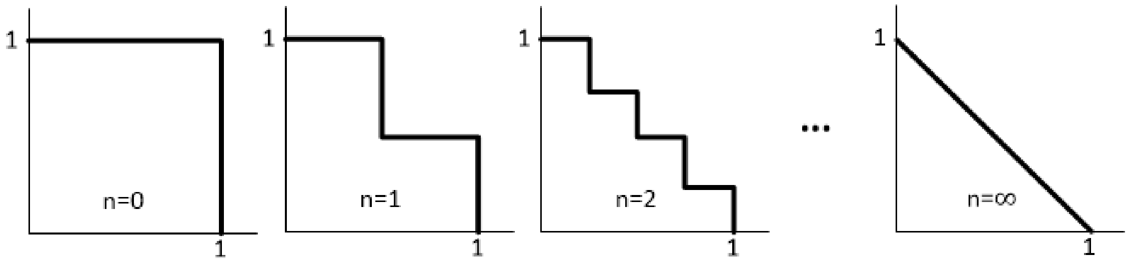
CUR: Самоподобные кривые

Тема занятия: Рекурсия

Рассмотрим итерационный процесс. Начнём с одной ступеньки. Длина этой кривой равна 2. Каждую сторону разобъём пополам и выгнем угол в обратную сторону, получим две ступеньки. Повторим разбиение, получим четыре, длина не меняется. Будем повторять до бесконечности, получим в пределе “прямую” линию длиной 2 вместо $\sqrt{2}$. Забавно, но противоречия нет: в отличие от настоящей прямой, у неё не существует производной ни в одной точке. Это разные объекты, хоть и похожи на первый (невооружённый) взгляд.



Бесконечно ломаная “лесенка”

Эта кривая — представитель (хоть и не самый красивый) семейства фрактальных кривых (кривых дробной размерности, находящихся где-то между одномерными и двумерными объектами), т.е. уже не линия, но ещё не фигура на плоскости. Они обладают свойством самоподобности, в данном случае любая её часть при $n = \infty$ идентична самой кривой с точностью до масштаба. И в случае n-ного прближения это в каком-то смысле верно, т.к. оно составлено из двух $(n - 1)$-ых приближений.

Кривая AB легко задаётся рекурсивно (обратите внимание: A, B, C, D — точки на плоскости, т.е. пары значений координат):

```
Curve(A, B, n) = Curve(A, C, n-1) + Curve(C, B, n-1),  где C = (A+B)/2,
Curve(A, B, 0) = Line(A, D) + Line(D, B),                  D = (Bx, Ay).
```

Сравните с рекурсивным определением факториала (произведения чисел от 1 до n)

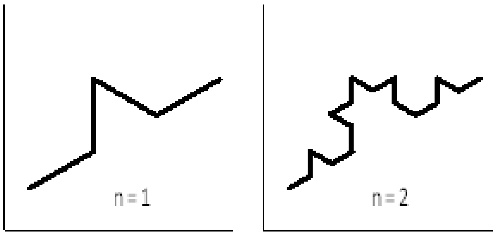
```
Factorial(n) = n * Factorial(n - 1),
Factorial(0) = 1,
```

которое практически один-в-один переносится на язык Си:

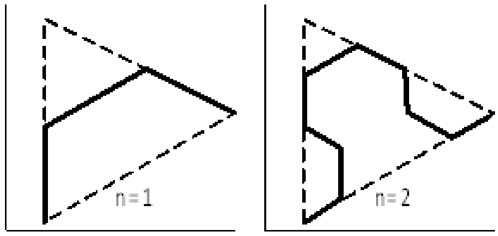
```
int factorial(int n)
{
  if (n == 0)
    return 1;

  return n * factorial(n - 1);
}
```

Есть и более интересные кривые, построение которых происходит аналогичным образом.



Кривая Коха



Кривая Серпинского

Построение других самоподобных кривых

1. **Постройте лесенку (+1 балл).** Напишите рекурсивную функцию рисования “лесенки” для заданного `n` по аналогии с рекурсивной функцией вычисления факториала. В отличие от математической функции, наша геометрическая рекурсивная функция (см. выше) должна:

   – для нулевого приближения найти угловую точку `D` и нарисовать отрезки прямых;

   – для ненулевого приближения найти промежуточную точку `C` и два раза нарисовать себя же в разных положениях и с меньшим приближением.

2. **Постройте кривую Коха (+1 балл).** Напишите ещё одну геометрическую рекурсивную функцию с чуть более сложными вычислениями. Основной трудностью в данном пункте является построение срединного перпендикуляра к отрезку, следует вспомнить векторную арифметику.

   – нулевое приближение является отрезком `AB`;

   – следующее приближение состоит из четырёх предыдущих (см. рис. при $n = 1$); отрезок разбивается на три равных части и из средней “вырастает шип” — равносторонний треугольник;

   – *подсказка:* два вектора перпендикулярны, если их скалярное произведение равно нулю; имея вектор `(x, y)`, можете ли вы придумать хоть какой-нибудь вектор `(u, v)`, чтобы $x \cdot u + y \cdot v = 0$?

3. **Добавьте интерактивность (+1 балл).** Оберните вызов функции рисования в цикл, ожидающий нажатия. В зависимости от нажатой клавиши, следует выполнить действие:

   – клавиша “плюс” увеличивает, “минус” уменьшает номер приближения и перерисовывает кривую; клавиша `ESC` завершает цикл и программу;

   – остальные клавиши игнорируются и не приводят к перерисовке кривой.

4. (*) **Постройте кривую Серпинского (+1 бонус)**. Нулевое приближение является отрезком. При переходе к следующему приближению каждый отрезок заменяется на трапецию, выгибающуюся то в одну, то в другую сторону (см. рис.). В этом пункте придётся мысленно достроить равносторонний треугольник, найдя его третью вершину `C` и середины сторон `D1`, `D2`, по которым уже организовать рекурсивный вызов.