

1. Опишите интерфейс IShape с методом double Area() и классы Square, Rectangle, Triangle, Circle, реализующие этот интерфейс.

Обратите внимание на правильную инициализацию и деинициализацию ресурсов.

2. Предложите исправление представленного ниже кода

```
FILE *f = fopen(name, "w");
```

*// Тут код, при котором может быть выброшено исключение*

```
fclose(f);
```

3. Предложите алгоритм для удаления дубликатов (или выбора уникальных элементов) из вектора. Оцените временную и пространственную сложность.

4. Сколько различных изображений можно получить из изображения NxN, если каждое следующее получается путем уменьшения предыдущего в 2 раза? В n раз?

5. Разработайте приложение с помощью фреймворка Qt, используя QWidgets.

Пирамида изображений - набор изображений, в котором каждое следующее изображение (слой пирамиды) получено путем уменьшения предыдущего в 2 раза.

Подробнее [https://en.wikipedia.org/wiki/Pyramid\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Pyramid_(image_processing))

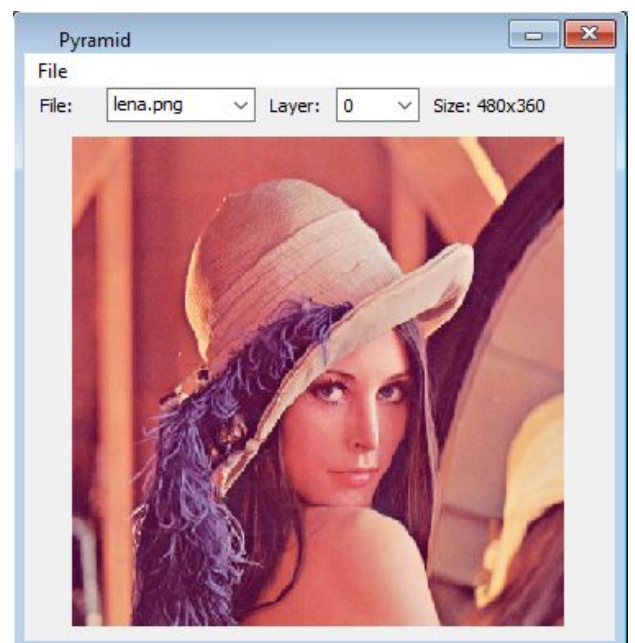
Приложение представляет собой просмотрщик такой пирамиды. Предлагается разработать поэтапно несколько версий программы:

1. Приложение позволяет выбрать файл изображения png или jpg из пункта меню и показывает его содержимое. Если размер изображения превышает размеры окна приложения, то должны появиться scrollbar-ы для перемещения по изображению.
2. При загрузке изображения по нему строится пирамида изображений.
3. Должна быть предусмотрена возможность произвольного переключения между слоями и оригинальным изображением с отображением номера слоя и его размера в пикселях. (т.е. слои можно "листать", например с помощью combobox). При отображении каждый слой должен быть увеличен до размера оригинального изображения (таким образом содержимое его "замылится").
4. Реализовать открытие нескольких файлов с возможностью их переключения. Отобразить список открытых файлов с помощью комбобокса. Список отсортировать в порядке увеличения диагонали изображения.
5. Снизить потребление памяти приложения (генерация слоя по требованию).
6. *Дополнительно:* позволить пользователю задавать произвольный коэффициент уменьшения слоя (например, не 2, а 1.6).

Справа представлен мокап одного из вариантов приложения, но следовать ему необязательно.

**Что хотелось бы увидеть в решении:**

- Умение разбираться с новыми фреймворками.
- Системы контроля версий, умение ими пользоваться (выделять функциональные изменения или улучшения в отдельные коммиты).
- ООП, обработка ошибок, нестандартных ситуаций (например, файл неподдерживаемого формата или неверные параметры командной строки).



Решение ожидается на языке C++. Возможно выложить решение на github и прислать ссылку на него.