# Задание 1

Создайте класс Point.

* Поля класса – координаты m\_x и m\_y.
* В конструкторе класса тело конструктора должно быть пустым!!! Используйте список инициализаторов.

Создайте класс Circle.

* Поля класса –
  + центр окружности (объект Point) m\_center;
  + радиус окружности (m\_radius);
  + «толщина» окружности (m\_thickness).
* Создайте конструктор с инициализацией всех параметров по умолчанию;

Создайте класс Cylinder – цилиндр, состоящий из нескольких окружностей.

Окружность 1

Окружность 2

Окружность 3

* Поля класса –
  + Основание цилиндра (указатель на объект класса Circle) (m\_pCircle);
  + Высота цилиндра (m\_height).
* В конструктор класса Cylinder передается только!!! N (число окружностей, из которых состоит цилиндр). В конструкторе создаем динамический массив из N объектов Circle (**создаются по умолчанию**) и, уже с помощью известных нам данных, инициализируем параметр m\_height.
* Создайте объект Cylinder
* Выведите на экран все окружности в формате



* Необходимо уменьшить высоту созданного цилиндра вдвое.

Подсказка: Для этого недостаточно просто разделить m\_height на 2. Также необходимо уменьшить в два раза массив объектов Circle. (потерей точности при округлении можно пренебречь)

Подсказка: Для уменьшения размеров динамического массива необходимо:

* Создать новый динамический массив с новым размером;
* Переписать почленно данные из старого в новый в соответствии с новым размером;
* Удалить старый массив (delete[] …);
* Направить старый указатель на новый массив.
* Выведите на экран все окружности в формате



# Задание 2

Создайте программу «Симулятор улья» /Hive simulator/

* Существуют матка (Queen) и рабочие пчелы (Worker).
* Матка командует четырьмя рабочими пчелами.
* Каждая пчела имеет вес (m\_weight);
* Каждая рабочая пчела знает 10 цветов (цветы для каждой пчелы одни и те же), который она может каждый день облетать для того, чтобы добыть необходимое количество меда.
* Каждый цветок может отдать за одно обращение только определенное количество меда.
* Каждый день матка отдает приказание каждой рабочей пчеле принести ей необходимое количество меда (одинаковое для каждой пчелы). В нашем примере – 50 единиц меда.
* Рабочая пчела, получив приказание, облетает в случайном порядке цветы до тех пор, пока не соберет необходимое количество меда.
* Благодаря случайному выбору два обращения подряд к одному цветку возможны, но не обязательны.
* Собрав мед, рабочая пчела должна сообщить матке, сколько обращений к цветам она сделала для выполнения задания.
* После этого она снова готова выполнять задание.

Продемонстрируйте работу программы «Симулятор улья».

* Подумайте, где можно применить при необходимости наследование, где использовать вложенные классы.
* При построении архитектуры классов используйте массивы указателей на объекты классов Worker (m\_workers) и Flower (m\_flowers).
* Логичным представляется следующий порядок: сначала создаем массив указателей Flower, затем массив указателей Worker, затем объект Queen
* В конструктор Flower передаем число, выбранное случайным образом, от 1 до 10, которое показывает, сколько меда за один раз может отдать цветок.
* Метод getHoney() (приказание матки рабочим пчелам) должен возвращать указатель на динамический массив, который хранит количество обращений к цветам, которое сделала каждая рабочая пчела/
* Выведите на экран содержимое динамического массива.
* Не забудьте корректно очистить память.

# Задание 3

Создайте класс Date (дата).

* Поля класса –
  + День (m\_day);
  + Месяц (m\_month)
  + Year (m\_year)
* По умолчанию полям присваивается дата 01.01.2000.

Создайте класс Employee.

* Поля класса –
  + Дата принятия сотрудника на работу (m\_hireDay);
  + Дата увольнения сотрудника (m\_lastDay);
  + Имя сотрудника – string (m\_name);
  + Заработная плата за месяц (m\_salary)
* В конструктор передается только имя и зарплата.
* Напишите функцию setHireAndLastDate, которая принимает на вход дату приёма на работу. В этой функции рассчитайте дату увольнения, зная, что срок службы работника составляет 100 дней, а приняли его
  + 23.12.2011г.
  + 23.12.2012г
* Необходимо учитывать возможность високосного года

Подсказка:

Простейшая функция обнаружения високосного года:

bool IsLeapYear = !(year % 4); //true, если високосный.

Подсказка:

Для работы с датами создайте локальный массив, хранящий количество дней в месяце.

int arr[2][12] = { { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 },

{ 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 } };

* Выведите обе даты на экран в формате:

Имя:

HireDate:

LastDate*:*