**Отчет по задаче «СВО»   
Шестаков Николай, БПМ-22-1**

*Ознакомиться с материалами по задаче вы можете по ссылке:*

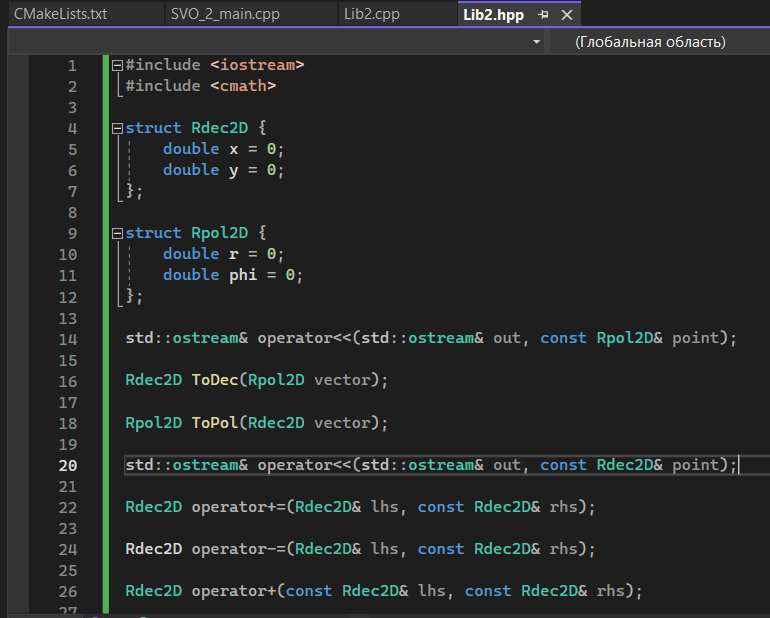
[*https://github.com/DR-Felix/DRFelix/tree/main/shestakov\_n\_a/SVO*](https://github.com/DR-Felix/DRFelix/tree/main/shestakov_n_a/SVO)

**Условие задачи:** В векторном виде даны координаты двух пушек, задана скорость Пушки 1 V\_1. В момент времени Ʈ = 0 первая пушка выстреливает Ядром 1 со скоростью V\_1. Через промежуток времени ∆t, за который звук выстрела первой пушки доходит до второй пушки, Пушка 2 должна выстрелить с такой скоростью V\_2, чтобы Ядро 2 столкнулось с Ядром 1 через время t, которое задает пользователь. При этом задается ограничение модуля скорости второй пушки – она не может стрелять быстрее заданной скорости.

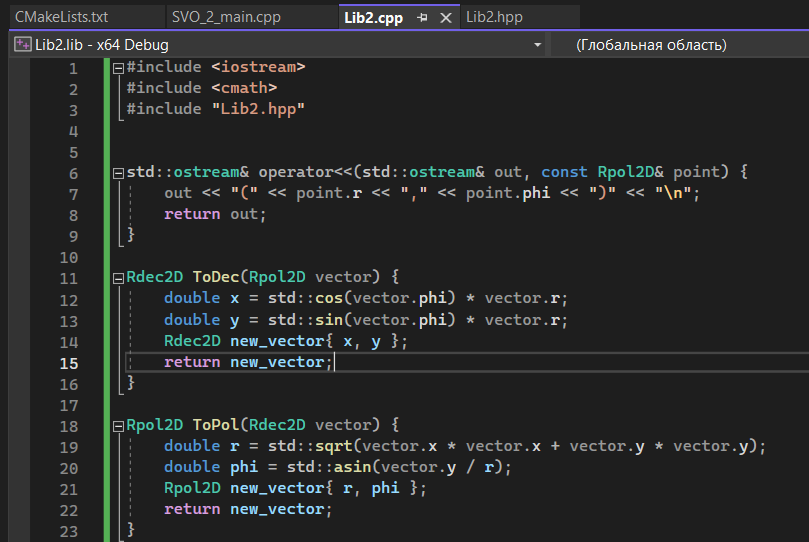
**Реализация:**

1. Для решения напишем библиотеку Lib, которая будет состоять из 2 файлов: Lib.hpp (который содержит объявление функций и структур) и Lib.cpp (который содержит реализацию функций и структур)
2. С помощью CMakeLists.txt свяжем библиотеку и проект SVO
3. Основной код представлен в файле SVO\_2\_main.cpp

**Lib.hpp**

Файл описывает объявление всех структур, а также перегруженных операторов для этих структур в полярном и декартовом векторном представлении.

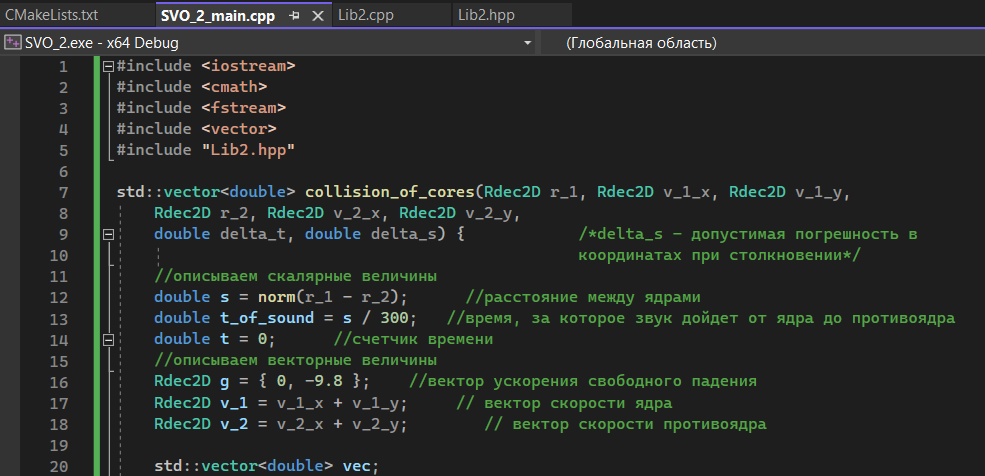
**Lib.cpp**

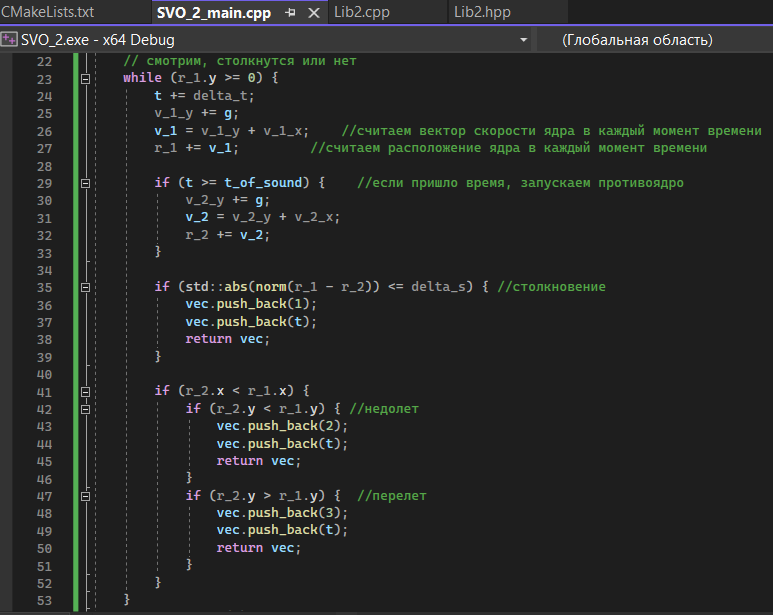
Файл содержит реализацию всех функций и структур, которые бы использовались в основном файле программы.

**Важно:** не забудьте подключить #include «Lib2.hpp»

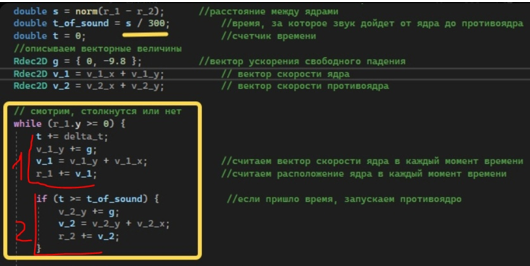
**SVO\_2\_main**

Программа состоит из большого количества логических блоков. Кратко дадим характеристику каждому из них:

**1 блок: «реализация»:**



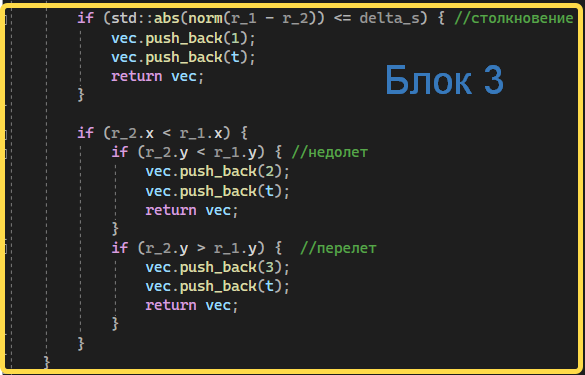
В данном блоке происходят основные расчеты: функция «collision\_of\_cores» отвечает за реализацию взаимодействия ядер: столкнуться или нет.  
На первом скриншоте закомментированы основные обозначения переменных.

На втором происходит расчетная часть. Суть заключается в работе с задачей с физической точки зрения: нам нужно использовать основные физические уравнения для баллистических задач:  


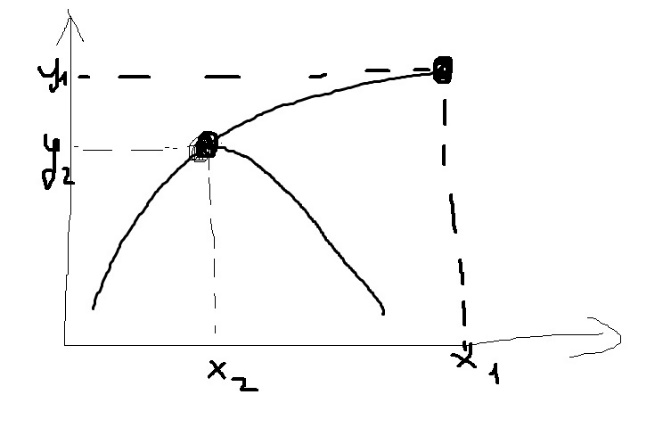
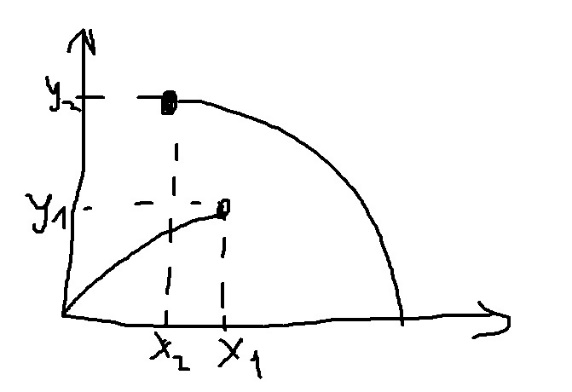
**Блок 1:**  
1) Меняем время на одну единицу  
2) Вектор скорости ядра v\_1 равен сумме векторов:  
v\_1\_x - вектор скорости по оси х, не меняется со временем  
v\_1\_y - вектор скорости по оси y, каждую единицу времени мы отнимаем вектор g - ускорение свободного падения, так как g действует только по оси y  
3) Присваиваем вектору скорости v\_1 новое значение (а оно у нас равно как раз v\_1\_x + v\_1\_y)  
4) Меняем расположение первого ядра (то есть прибавив к r\_1 v\_1)

**Блок 2:**Мы ждем момента, когда звук долетит до второго ядра.  
Как только этот момент наступает, мы начинаем делать те же действия со вторым ядром.

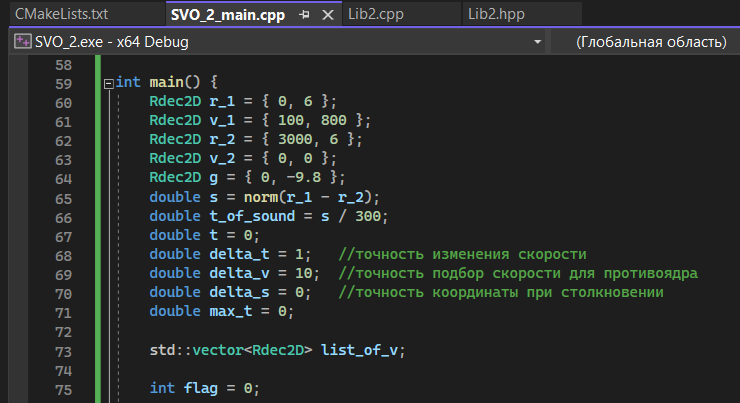
**Блок 3:**

****

В данном случае происходит своеобразный подсчет результатов. Если norm(r\_1-r\_2) <= delta\_s, то есть если модуль расстояния между ядрами входит в допустимую погрешность, возвращаем 1, в массив vec фиксируем 1 (флажок столкновения) и время столкновения.

В противном случае помещаем в vec «2» для недолета и время недолета или «3» для перелета и время перелета (см. рисунки ниже: первый для перелета, второй – для недолета): 

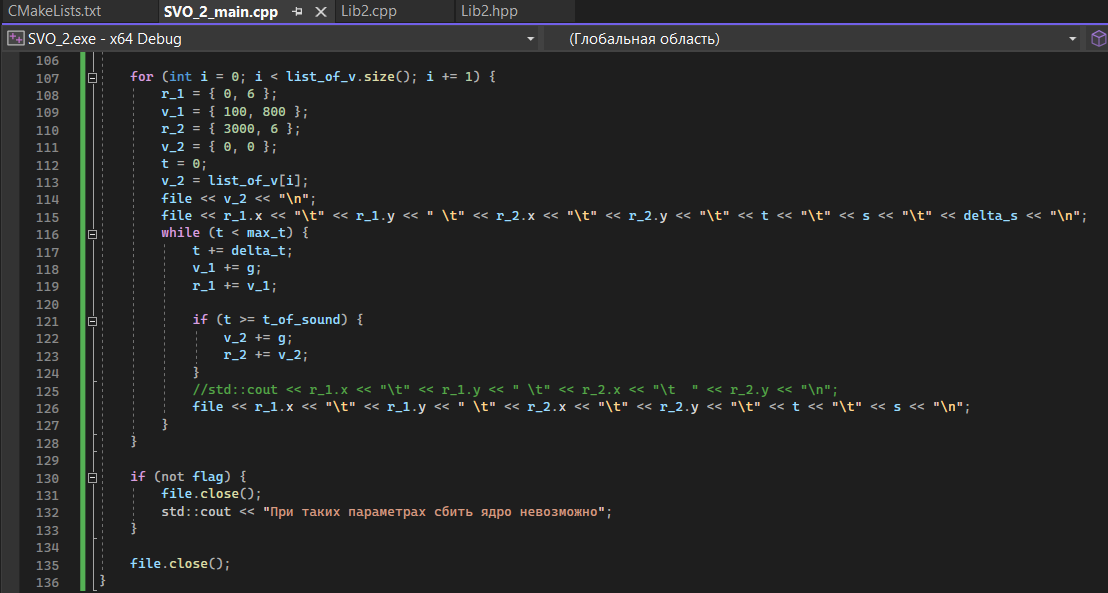
Переходим в основной код, **блок 4:**



Объявляем все переменные, дельты и массив list\_of\_v.

В **блоке 5:** Изображение выглядит как текст, снимок экрана, экран

Автоматически созданное описание

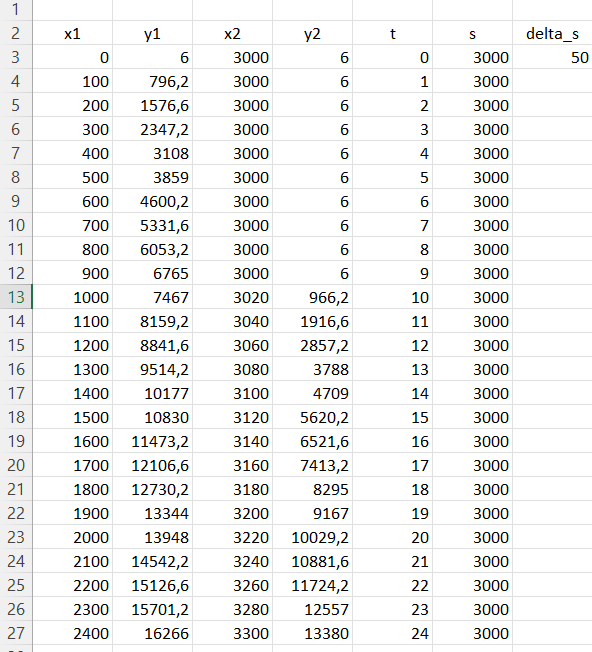
Здесь происходит заполнение vec благодаря функции, разобранной в блоках 1 и 2 и вывод в 85 строке значение v\_2 при заданных параметрах (которые можно менять).  
  


В финальном блоке мы фиксируем все переменные в каждый момент времени в файл coords3.csv, который автоматически создает Сmake при сборке.

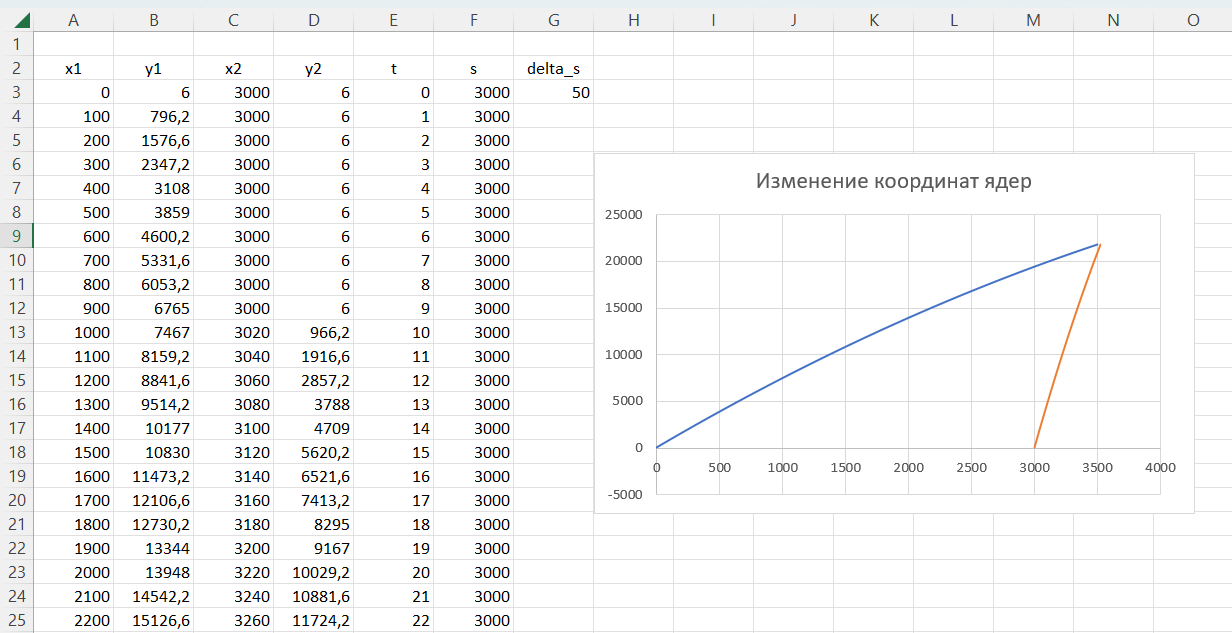
Итого: программа предусматривает все возможные конфигурации стартовых значений и в зависимости от них будет выводить найденное значение v\_2 в векторном виде.

**Excel**

Изначально мы получаем все эти столбики в одном столбце из-за особенностей вывода в файл. Перейдем в Данные -> текст по столбцам -> знак табуляции и разделим все значения до такого вида:



Как вы видите, в таблице представлены координаты первой и второй пушек x1, y1 и x2, y2 соответственно в момент времени t и стартовое расстояние между ядрами s до момента времени s = 0 до момента столкновения. Дальше расчеты уже не ведутся.

Для большей визуализации создадим график, который бы иллюстрировал ход ядер: 

В зависимости от стартовых значений график также будет меняться вместе с координатами и прочими характеристиками. Задача решена.