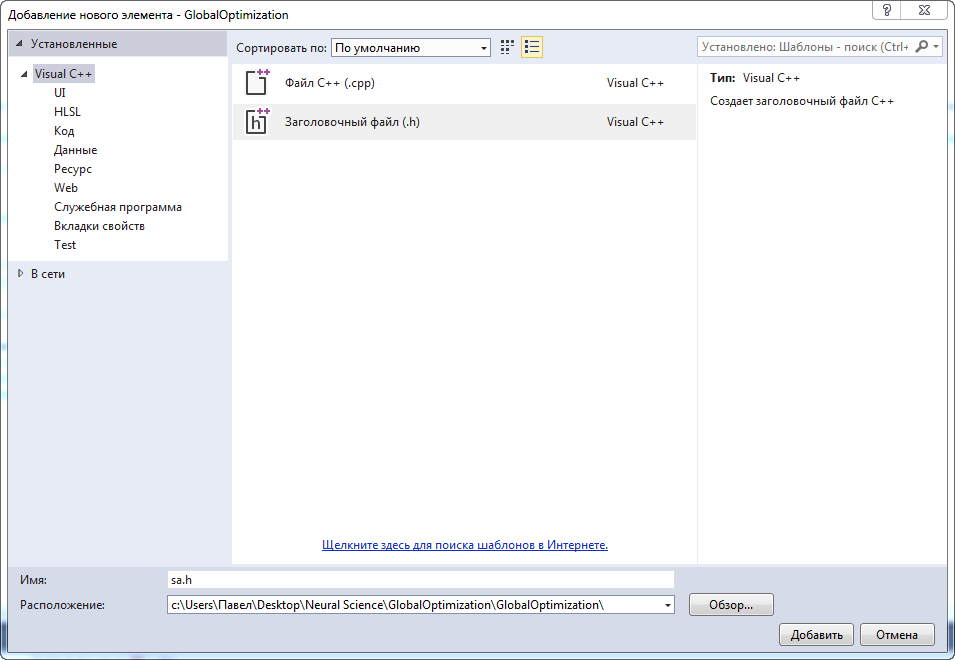
**Методическое пособие по разработке методов**

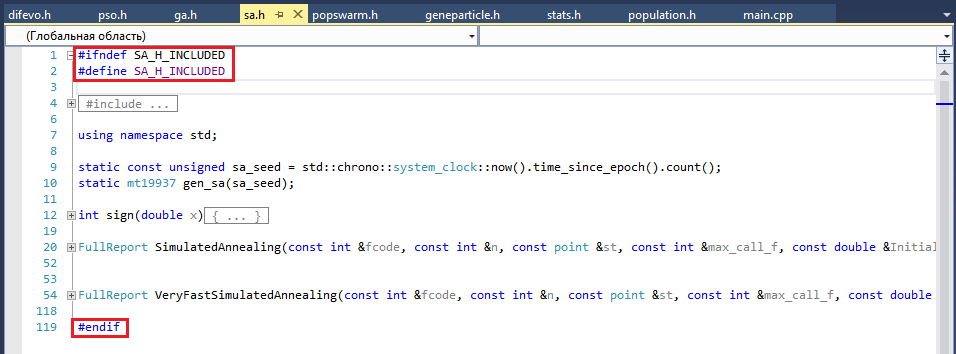
Для того, чтобы разработать новый метод, необходимо сделать следующее:

1. Создайте нужное количество заголовочных файлов (.h) для выбранного метода. Имена должны кратко отображать суть. Например, имитация отжига Simulated Annealing - sa.h. В нем (них) будет реализован непосредственно сам метод.



1. Теперь необходимо прописать программный код.

В самом начале - стражи включения #ifndef и #define, после которых название заголовочного файла + «\_H\_INCLUDED», в конце – закрытие.

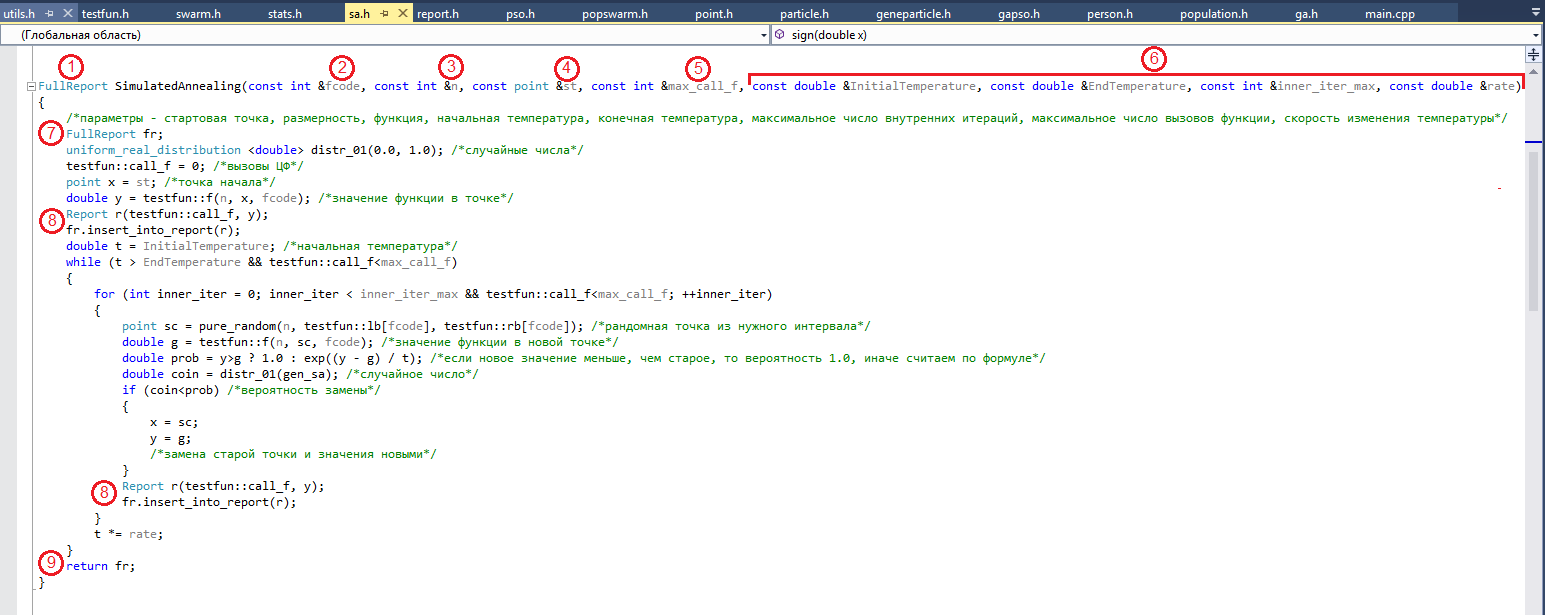


Дальше сам метод. Он должен быть реализован в виде функции, которая возвращает объект класса FullReport (1). Эта функция имеет следующие обязательные параметры:

* порядковый номер целевой функции (2);
* размерность задачи (3);
* стартовая точка типа point (для одноточечных методов) или вектор стартовых точек (для многоточечных методов) (4);
* максимальное число вызовов целевой функции (5).

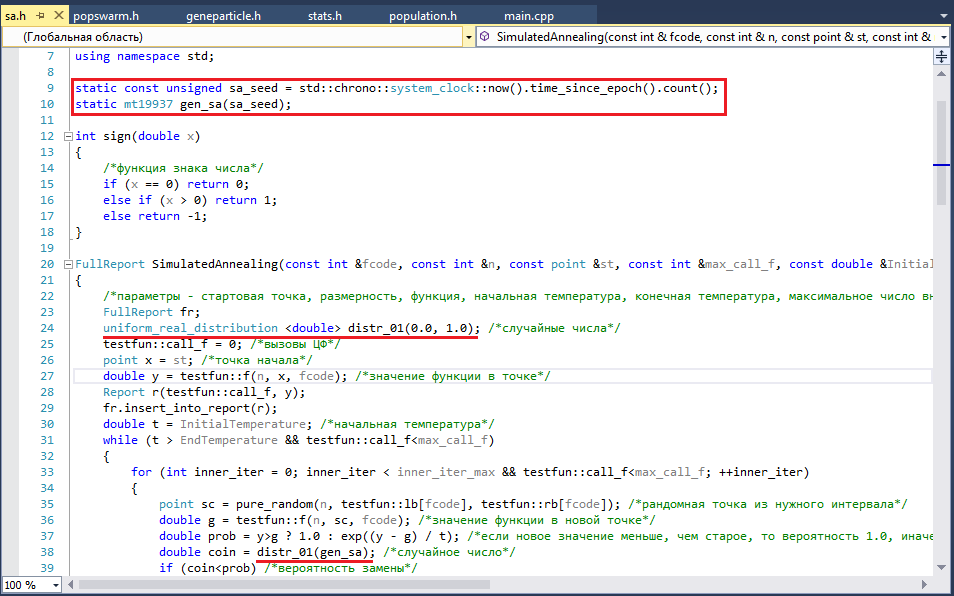
Остальные параметры определяются разработчиком метода (6). Все параметры могут быть расположены в произвольном порядке. Рекомендуется передавать параметры по ссылке (&) и указывать перед типом const.

Тело функции начинается с создания объекта класса FullReport (7) – он содержит информацию для отчетности по всему одинарному запуску, такую как количество вызовов тестовой функции и её значение на каждом шаге. Также создается объект класса Report – туда записываются данные по одной итерации. Добавление его в полный отчет производится с помощью метода insert\_into\_report (8). Функция разрабатываемого метода возвращает объект класса FullReport (9), созданный в начале.

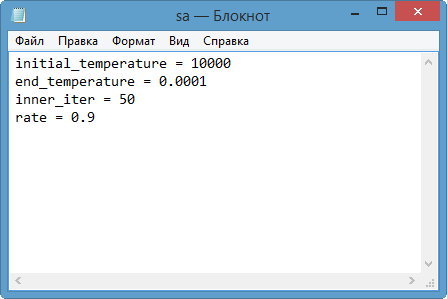


**Примечание:**

Рекомендуем использовать генератор псевдослучайных чисел из библиотеки <random.h> (например, Вихрь Мерсенна mt19937).

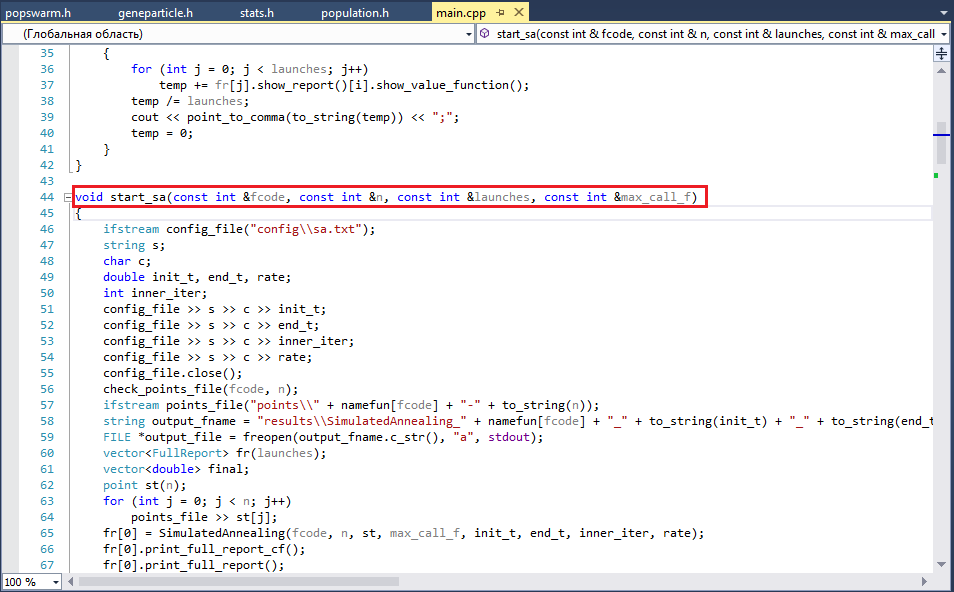


1. В папку config добавьте конфигурационный файл формата txt реализованного метода с соответствующим названием. В него пропишите все параметры алгоритма в формате: название + « = » + значение.

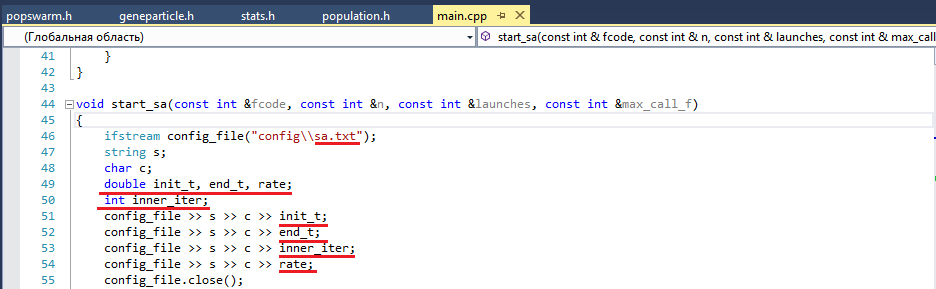


1. В main.cpp добавьте функцию. Имя функции состоит из «start\_» + название метода. Тип возвращаемого функцией значения – void. Эта функция имеет следующие параметры:

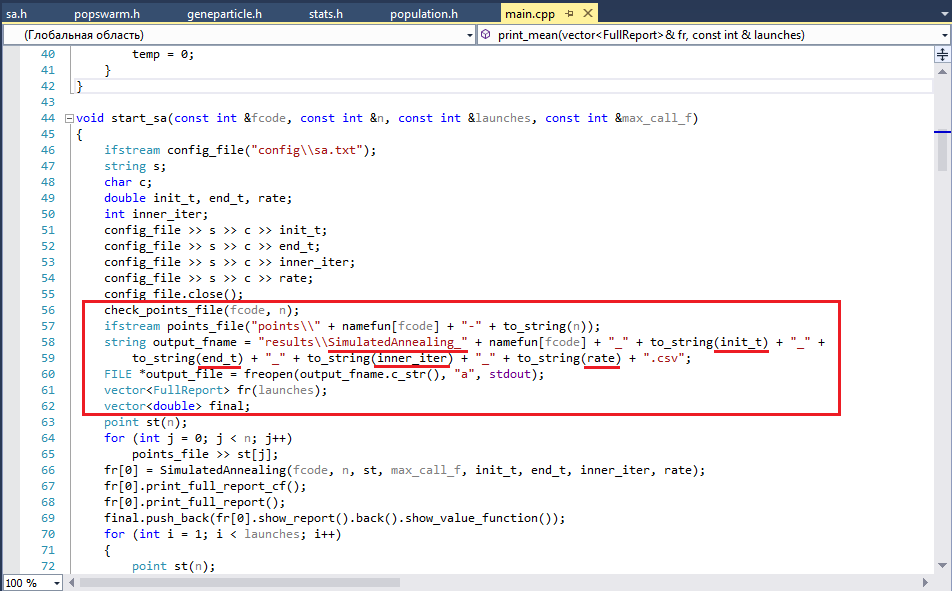
* порядковый номер целевой функции;
* размерность задачи;
* число запусков для усреднения;
* максимальное число вызовов целевой функции.



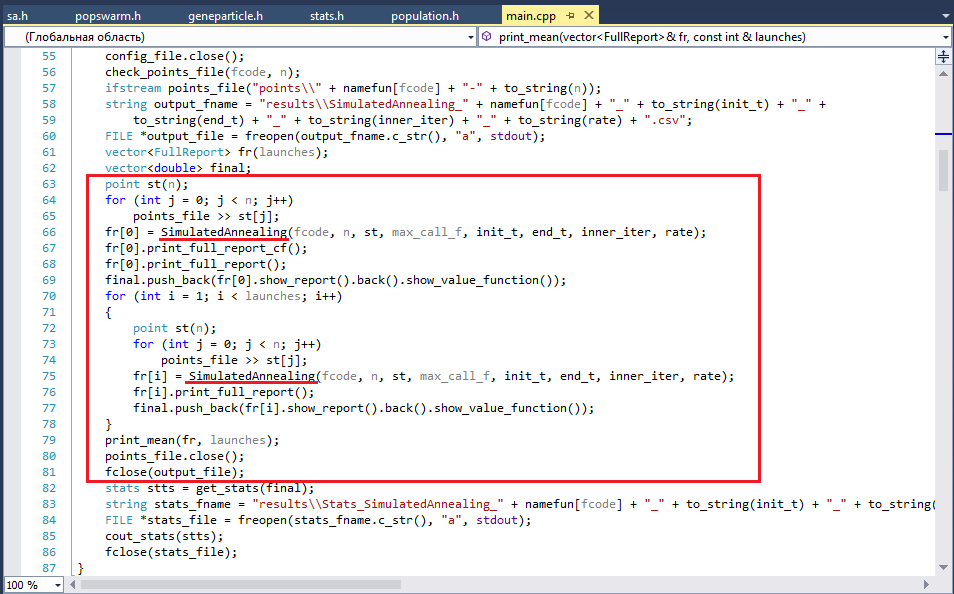
Сначала идёт считывание значения параметров метода из конфигурационного файла.



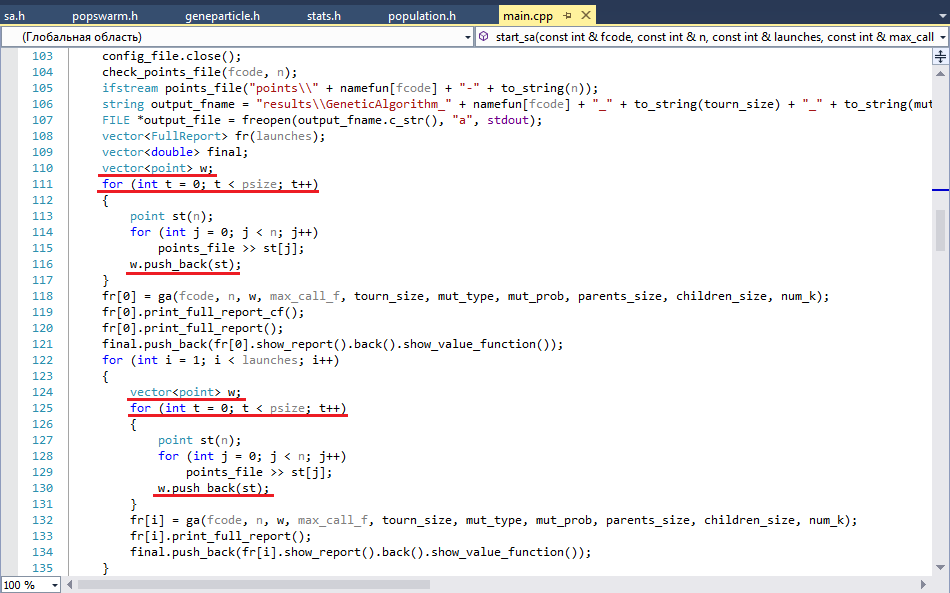
Затем идёт проверка наличия файла стартовых точек для выбранного метода и функции. В случае отсутствия, он будет сгенерирован и сохранен автоматически. Сразу же создается файл результатов.



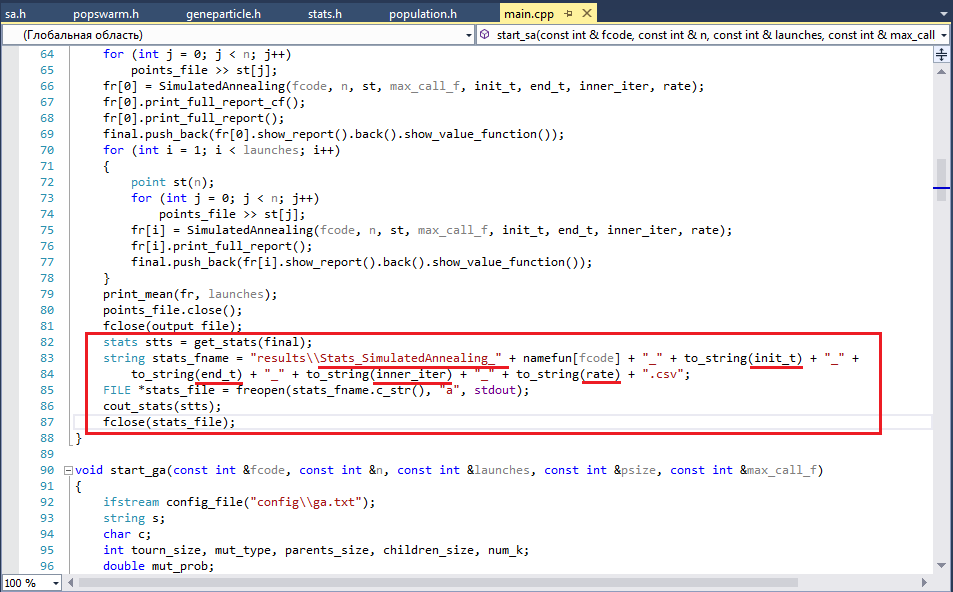
Дальше считываются стартовые точки с соответствующего файла и вызывается сама функция метода. Данные для отчета заносятся в файл результатов.



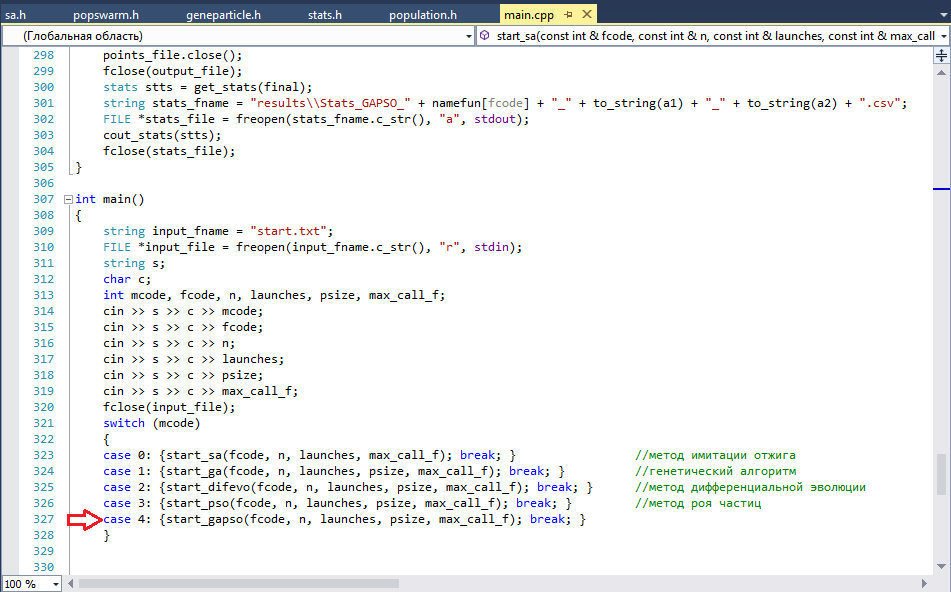
Если метод многоточечный:



Статистика считается по завершению.



1. Добавьте вызов функции в main().



1. В main.cpp подключить заголовочный файл метода.

