

# №6 Работа с методами объектов

Вам даны:

1. Файл `geometry.hpp` — содержит определения необходимых типов и структур:
  - Обёртка `Double`, хранящая числовое значение и предоставляющая вспомогательные методы сравнения (`is_equal`, `is_less`), чтобы все вещественные сравнения были надёжными и согласованными.
  - Перечисление `Kind` для различия трёх типов геометрических объектов: прямая (`Line`), луч (`Ray`), отрезок (`Segment`).
  - Структура `Point2D`, содержащая две координаты типа `Double` и методы для базовых вектороподобных операций (сложение, вычитание), а также метод вычисления расстояния до другой точки.
  - Структура `Geometry`, объединяющая `Kind` с двумя значениями `Point2D`, называемыми `begin` и `end`, чтобы представлять геометрический объект.
2. Файл `main.cpp` — содержит тесты и использует `Double::is_equal` для сравнения ожидаемых и вычисленных расстояний (этот файл изменять нельзя!).

## Задача

### Note

Разрешается добавлять любые методы/функции в `geometry.hpp` (в т.ч. вспомогательные для векторов, проекций, расстояний точка ↔ прямая/луч/отрезок и т.д.).

### Important

Файл `main.cpp` менять не требуется, только `geometry.hpp`.

Требуется реализовать функцию `Double distance(Geometry g1, Geometry g2)` — расчёт минимального евклидова расстояния между двумя объектов типа `Geometry`.

Пара подсказок:

- Расстояние между двумя объектами — минимальное расстояние между любой парой точек по одному на каждом объекте. Если объекты пересекаются — расстояние = 0.
- Для сравнений используйте `Double` и его методы `Double::is_equal` / `Double::is_less`.

## Критерии оценивания

Всего тестов 60 штук. Тесты разделены на 6 равных категорий по типам пар геометрий:

1. Line — Line
2. Line — Segment
3. Line — Ray
4. Segment — Segment
5. Segment — Ray

## 6. Ray — Ray

Каждая категория содержит по 2 подкатегории (каждая по 5 тестов):

- пересекающиеся случаи (ожидается расстояние 0.0)
- непересекающиеся случаи (минимум достигается на проекции, на границе отрезка/луча или в случае параллельности)

Частичные баллы возможны: так как тесты группируются по категориям/подкатегориям, успешное прохождение тестов даёт прирост счётчика пройденных тестов. Пороговое значение для получения минимального зачёта установлено в 30 тестов (если всегда возвращать 0.0).

**Формула для вычисления итогового балла:**

$$\text{score} = \begin{cases} 0, & \text{if } \text{passed} \leq \text{threshold}, \\ 100\% \times \frac{\text{passed} - \text{threshold}}{\text{total} - \text{threshold}}, & \text{otherwise.} \end{cases}$$

Символ `passed` обозначает число пройденных тестов, `total` — общее количество тестов (60), а `threshold` — пороговое значение (30).

Если `passed` не превышает порог, результат равен нулю. В противном случае дробная формула линейно масштабирует избыточное количество (`passed` – `threshold`) относительно числа ненулевых тестов (`total` – `threshold`) и в результате получаем `score` в диапазоне от 0% до 100%.

## Примечания и советы

---

- Сначала реализуйте и протестируйте функции расстояния от точки до прямой/луча/отрезка с проверкой на простых случаях.
- Затем реализуйте общую логику для двух объектов: решите несложную "Line2Line" задачу, после чего добавьте отсечение по параметрам для Ray/Segment и рассмотрите расстояние до их границ.
- Аккуратно обрабатывайте численные ошибки, нулевые векторы и параллельность.