

# Домашнее задание

Всего баллов:

10

## 1. Points

*Это задание не на динамический полиморфизм, а на общую работу с Trait-ами.*

Баллы:

2

Вам дана структура Point, хранящая в себе две координаты, `x` и `y` производного типа `<T>` (в общем случае оно не обязано быть числом).

Требуется реализовать для этой структуры сложение, вычитание (покомпонентное), скалярное произведение, скалярный квадрат для тех `T`, которые допускают такие операции.

## 2. 2D рейтрейсер.

Дано несколько типов 2D фигур.

Все фигуры должны реализовать `trait Intersectable`, для которого требуется предоставить функцию, возвращающую глубину и цвет в заданной точке (или `None`, если вне фигуры).

Фигур четыре:

- **Круг**, задается центром и радиусом.
- **Квадрат**, задается левым нижним и правым верхним углом. Гарантируется, что точки в нём будут в нужном порядке (например, в тестах не будет передаваться вместо этого, например, левая верхняя и нижняя правая).
- **Треугольник**. *Оценивается дополнительно.* Так как алгоритм определения того, лежит ли точка внутри треугольника может быть не очевиден, он приведен далее в **Дополнении**.
- **Background**. По сути — это заливка всей картинки заданным цветом на заданной глубине.

Баллы:

3

Границы фигур *не включаются*.

У каждой фигуры есть «глубина», целое число. При пересечении отрисовывается фигура с *наименьшей* глубиной. Если глубины совпадают, последняя по порядку фигура в массиве должна отрисовываться ближе.

Далее нужно реализовать функцию, которая по массиву фигур и точке возвращает *цвет* в этой точке. Если здесь *нет* фигур, нужно вернуть чёрный, *цвет по умолчанию*.

Как вы должны были понять из лекций, есть несколько способов передать массив фигур с заданными свойствами.

### 2.1. Отрисовка из EnumContext

Массив на входе содержит *алгебраический тип данных* Figure, содержащий в себе все перечисленные фигуры.

Баллы:

2

### 2.2. Отрисовка из DynContext

Баллы:

3

После этого задания должен наступить катарсис — работать с `dup` на самом деле проще, чем с `enum`.

Массив на входе содержит элементы, динамически реализующие `trait Intersectable`. Вообще говоря, это могут быть произвольные элементы, для которых реализовали такой `trait` — например, если использовать этот файл как библиотеку, пользователь может реализовать `trait` для любого элемента, например, градиентно раскрашенной звездочки с меняющейся глубиной и передать в эту функцию.<sup>1</sup>

### 2.3. Красивые картинки

Потестировать ваш замечательный raytracer можно с помощью функции `draw_image` (пример см. в функции `main`). Он будет выводить красивую картинку в корень проекта.

Картинки имеют разрешение 500×500, `(0, 0)` находится по центру.

Бесплатно —  
сколько  
влезет!

## Дополнение

Алгоритм проверки точки в треугольнике:

Кратко — см. [StackOverflow](#).

# ! TODO !

Знающий человек вроде Вовы или Ильи сделает

---

<sup>1</sup>Тестировать, правда, мы будем только на вышеперечисленных элементах с вашими реализациями.