

Рябов Ю.В. < riabovvv@gmail.com >

Санкт-Петербургский научно-исследовательский центр  
экологической безопасности РАН

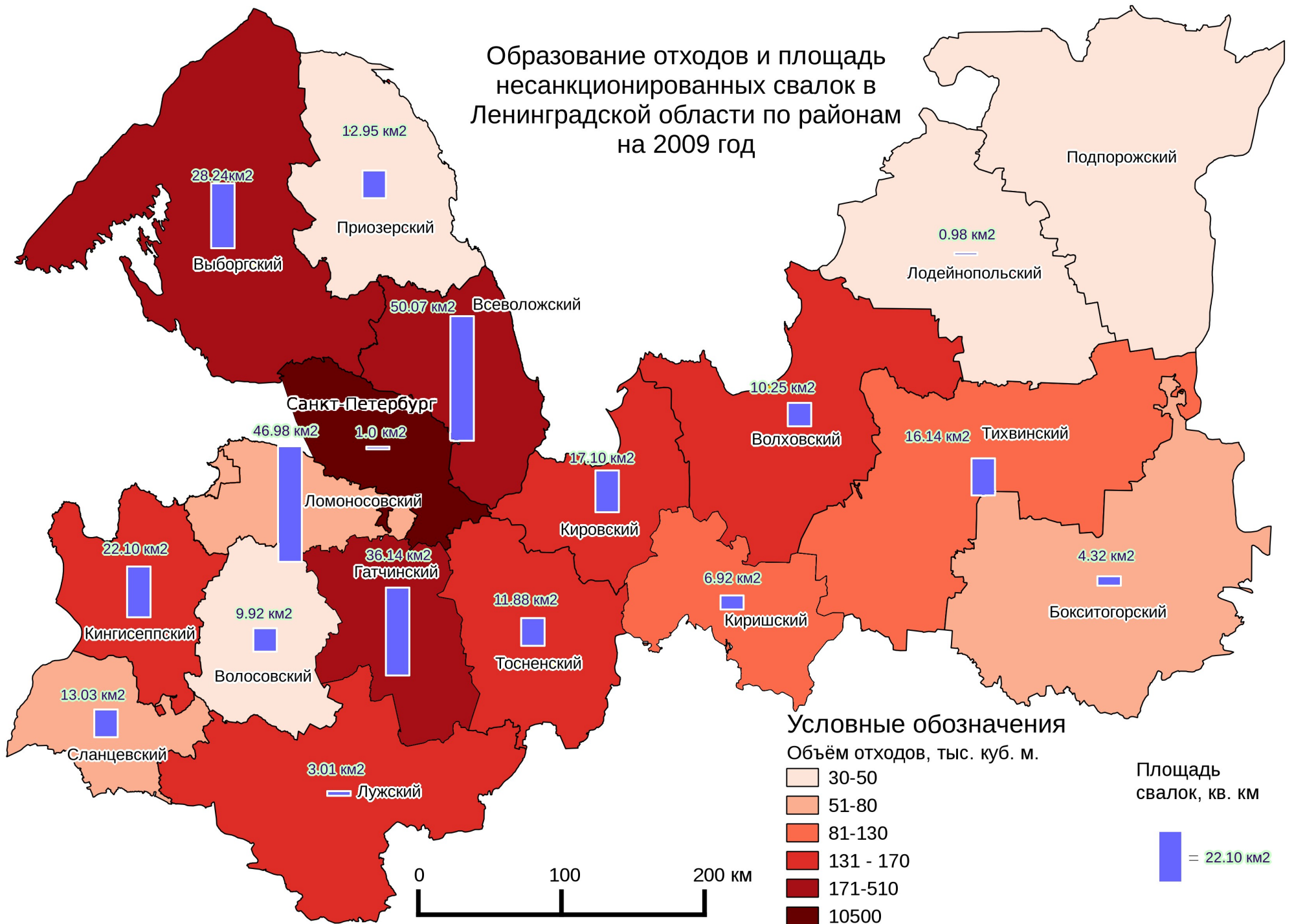
*Разработка алгоритма определения  
ширины полигона в заданном  
направлении и его реализация при  
помощи Python и QGIS API*

17-18 ноября 2012 г.  
Москва

# Мотивация

Вначале была карта...

# Образование отходов и площадь несанкционированных свалок в Ленинградской области по районам на 2009 год



# Мотивация

Затем было обсуждение способа нахождения максимальной длины полигона в заданном направлении на [gis.stackexchange.com](https://gis.stackexchange.com).

# Области применения

- Для справочных целей (протяжённость Байкала с севера на юг).
- Для анализа некоторых типов пространственно агрегированных данных.
- ...???

# Алгоритмы

- Сканирующая линия с равными интервалами
- Сканирующая линия в узлах
- Растеризация (???)

# Сканирующая линия с равными интервалами

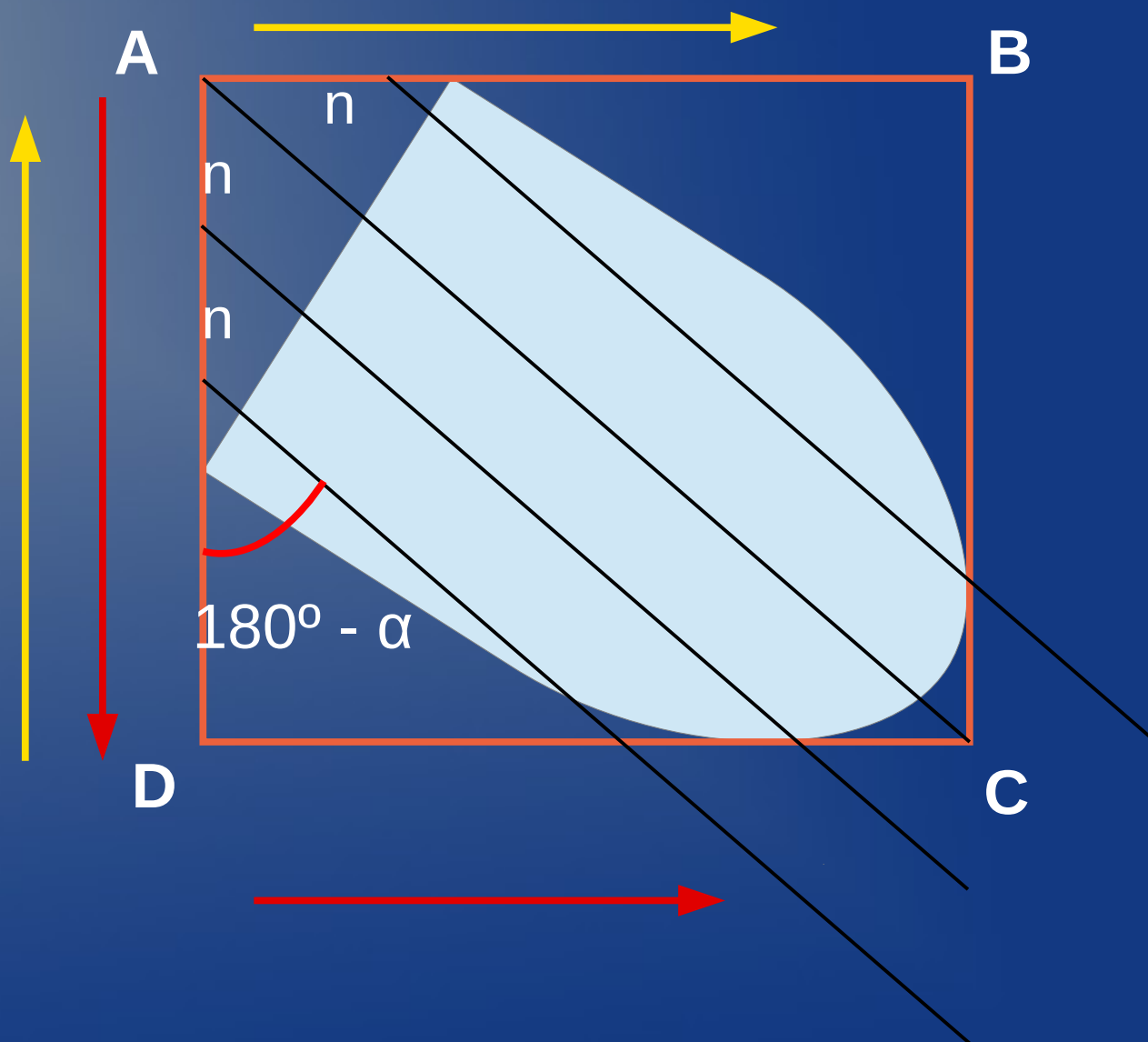
ABCD — минимальный описывающий прямоугольник.

$|AC|$  — максимально возможная ширина геометрической фигуры.

$\alpha$  — азимут.

$n$  — интервал сканирования.

$\rightarrow$  — направления сканирования (ADC или DAB). Зависят от величины  $\alpha$ .



# Сканирующая линия в узлах

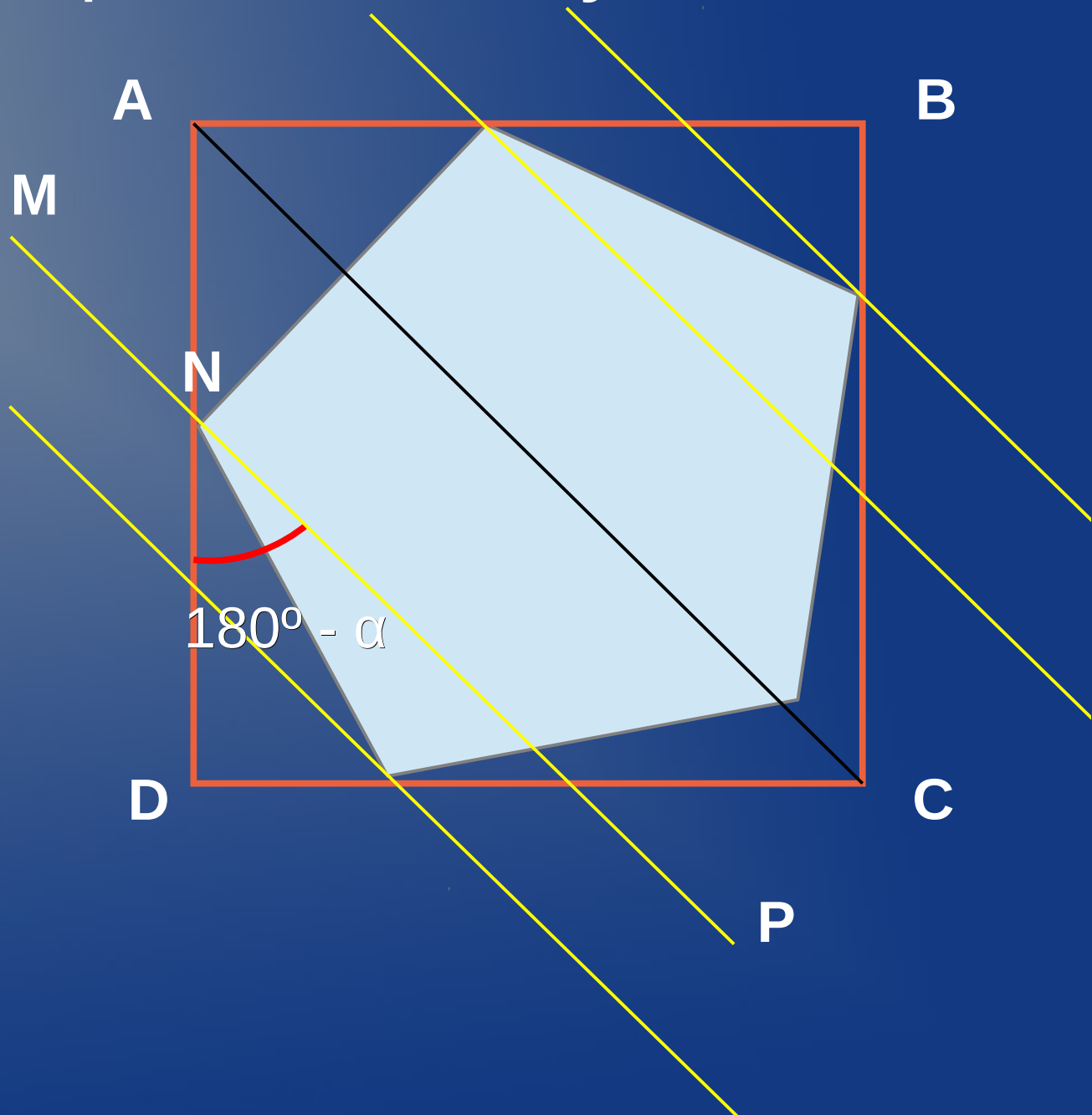
ABCD — минимальный описывающий прямоугольник.

$|AC|$  — максимально возможная ширина геометрической фигуры.

$\alpha$  — азимут.

MP — сканирующая линия

$|MN| = |NP| = |AC|$





# Почему реализация на Python + QGIS API?

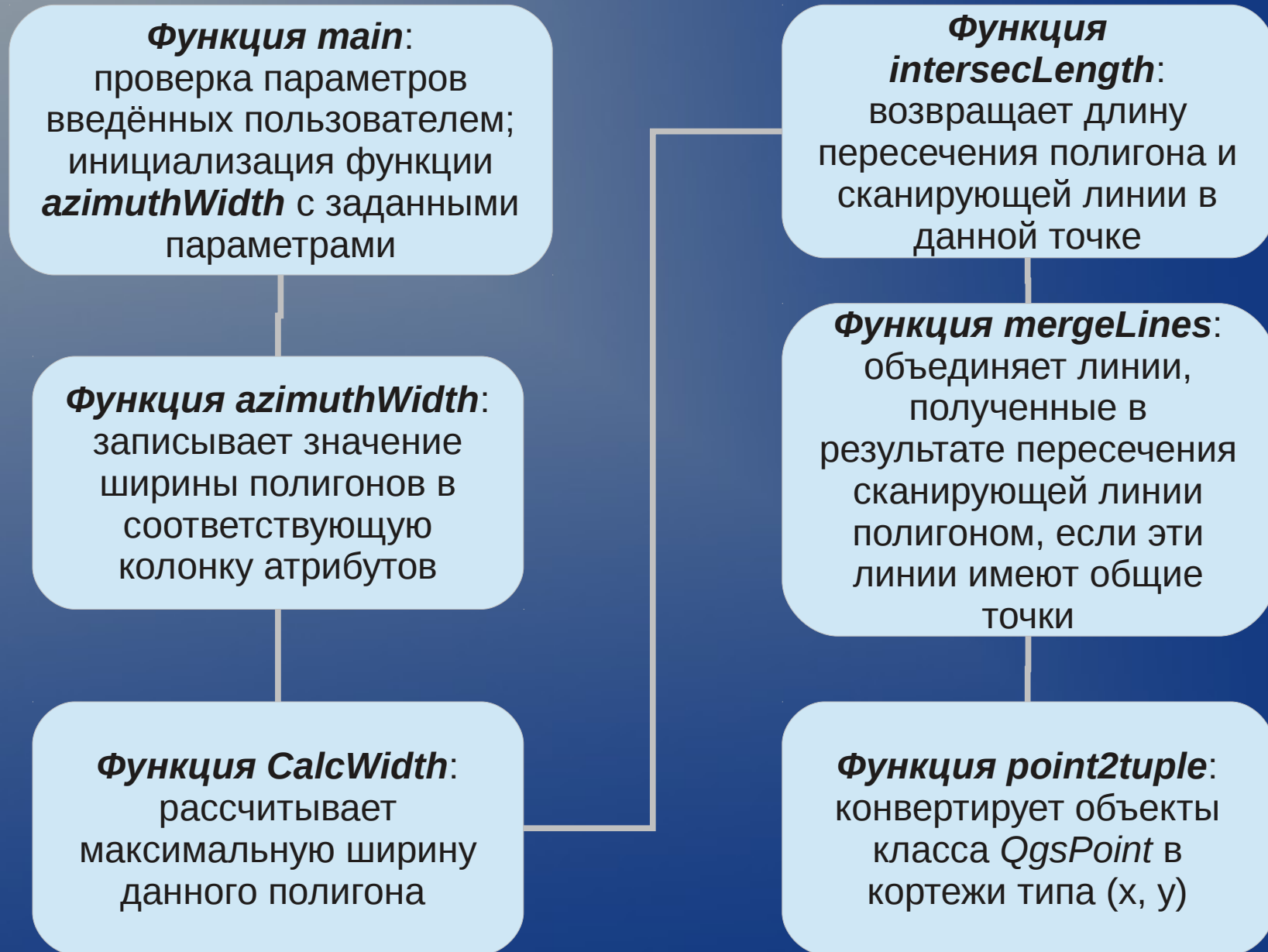
## Почему:

- На Python очень легко писать скрипты;
- Python уже предустановлен на всех моих машинах (ибо везде Linux);
- QGIS тоже на всех машинах;

## Зачем:

- Самообучение использованию QGIS API;
- Получение навыков создания собственного Open Source приложения (включая написание документации и разворачивание репозитория);
- Помощь нуждающимся в подобном приложении.

# Реализация в Python + QGIS API



# Реализация в Python + QGIS API



**Функция *main*:**  
проверка параметров  
введённых пользователем;  
инициализация функции  
***azimuthWidth*** с заданными  
параметрами

**Функция *azimuthWidth*:**  
записывает значение  
ширины полигонов в  
соответствующую  
колонку атрибутов

**Функция *CalcWidth*:**  
рассчитывает  
максимальную ширину  
данного полигона

**Функция *intersecLength*:**  
возвращает длину  
пересечения полигона и  
сканирующей линии в  
данной точке

**Функция *mergeLines*:**  
объединяет линии,  
полученные в  
результате пересечения  
сканирующей линии  
полигоном, если эти  
линии имеют общие  
точки

**Функция *point2tuple*:**  
конвертирует объекты  
класса *QgsPoint* в  
кортежи типа (x, y)

# Реализация в Python + QGIS API



**Функция *main*:**  
проверка параметров  
введённых пользователем;  
инициализация функции  
***azimuthWidth*** с заданными  
параметрами

В консоли пользователь вводит следующие параметры:

- Название shp-файла;
- Название поля для хранения значений;
- Выбирает минимальное или максимальное значение ширины будет рассчитываться (max/min);
- Как будет определяться ширина для полигонов со сложной геометрией (abs/rel) (подробнее на следующем слайде...);
- Алгоритм расчёта: byStep, byVertex, mix;
- Шаг сканирования для алгоритмов byStep и mix.

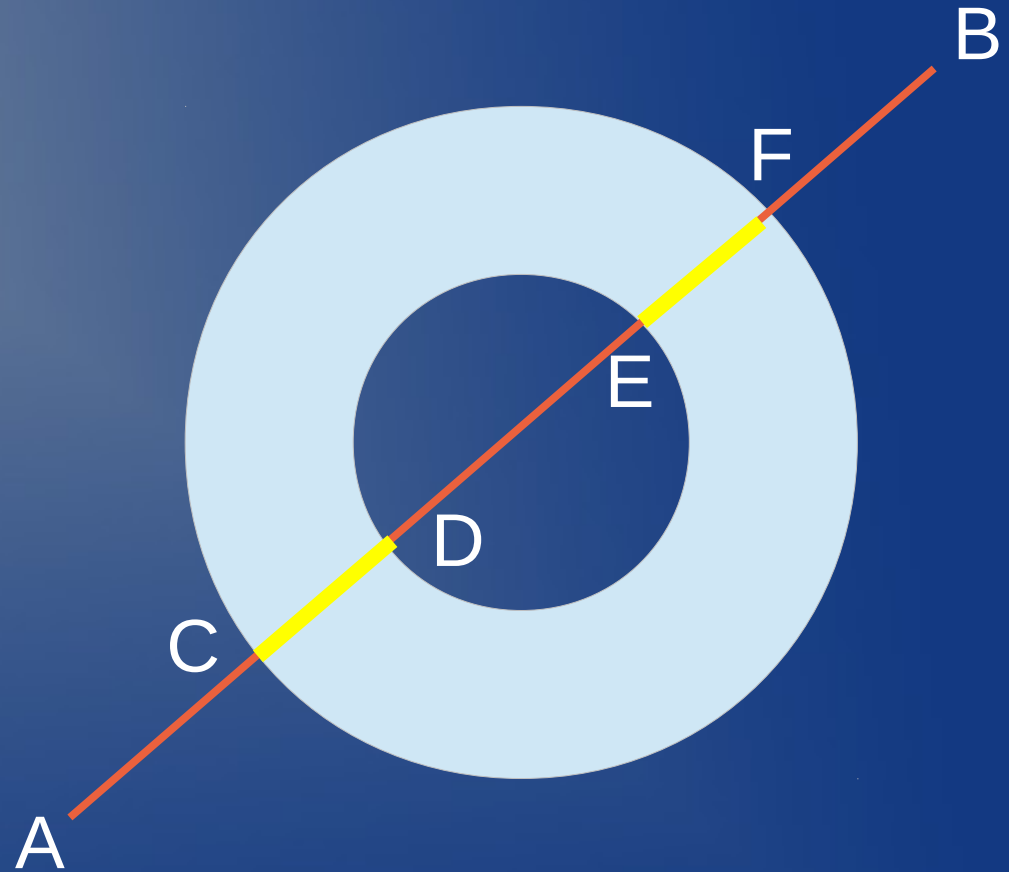
# Реализация в Python + QGIS API

Пример сложной геометрии:

AB — сканирующая линия.


CD и EF — результаты пересечения сканирующей линии полигоном.

Пользователь задаёт, хочет ли он получить в качестве результата наибольшее (наименьшее) значение пересечений (  $|CD|$  или  $|EF|$  ), либо их сумму (  $|CD| + |EF|$  )



# Реализация в Python + QGIS API

**Функция *main*:**  
проверка параметров  
введённых пользователем;  
инициализация функции  
***azimuthWidth*** с заданными  
параметрами



**Функция *azimuthWidth*:**  
записывает значение  
ширины полигонов в  
соответствующую  
колонку атрибутов

**Функция *CalcWidth*:**  
рассчитывает  
максимальную /  
минимальную ширину  
данного полигона

**Функция *intersecLength*:**  
возвращает длину  
пересечения полигона и  
сканирующей линии в  
данной точке

**Функция *mergeLines*:**  
объединяет линии,  
полученные в  
результате пересечения  
сканирующей линии  
полигоном, если эти  
линии имеют общие  
точки

**Функция *point2tuple*:**  
конвертирует объекты  
класса *QgsPoint* в  
кортежи типа (x, y)

# Реализация в Python + QGIS API

**Функция *main*:**  
проверка параметров  
введённых пользователем;  
инициализация функции  
***azimuthWidth*** с заданными  
параметрами

**Функция *azimuthWidth*:**  
записывает значение  
ширины полигонов в  
соответствующую  
колонку атрибутов

**Функция *CalcWidth*:**  
рассчитывает  
максимальную /  
минимальную ширину  
данного полигона

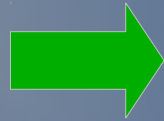


**Функция *intersecLength*:**  
возвращает длину  
пересечения полигона и  
сканирующей линии в  
данной точке

**Функция *mergeLines*:**  
объединяет линии,  
полученные в  
результате пересечения  
сканирующей линии  
полигоном, если эти  
линии имеют общие  
точки

**Функция *point2tuple*:**  
конвертирует объекты  
класса *QgsPoint* в  
кортежи типа (x, y)

# Реализация в Python + QGIS API



**Функция *CalcWidth*:**  
рассчитывает  
максимальную /  
минимальную ширину  
данного полигона

- Если выбран алгоритм *byStep* или *mix* и указанный шаг больше чем 0.01 от длины наименьшей стороны минимального описывающего прямоугольника для данного полигона, то принудительно используется шаг в 0.01 длины наименьшей стороны минимального описывающего прямоугольника.



# Реализация в Python + QGIS API

**Функция *main*:**  
проверка параметров  
введённых пользователем;  
инициализация функции  
***azimuthWidth*** с заданными  
параметрами

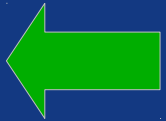
**Функция *azimuthWidth*:**  
записывает значение  
ширины полигонов в  
соответствующую  
колонку атрибутов

**Функция *CalcWidth*:**  
рассчитывает  
максимальную /  
минимальную ширину  
данного полигона

**Функция *intersecLength*:**  
возвращает длину  
пересечения полигона и  
сканирующей линии в  
данной точке

**Функция *mergeLines*:**  
объединяет линии,  
полученные в  
результате пересечения  
сканирующей линии  
полигоном, если эти  
линии имеют общие  
точки

**Функция *point2tuple*:**  
конвертирует объекты  
класса *QgsPoint* в  
кортежи типа (x, y)



# Реализация в Python + QGIS API

**Функция *main*:**  
проверка параметров  
введённых пользователем;  
инициализация функции  
***azimuthWidth*** с заданными  
параметрами

**Функция *azimuthWidth*:**  
записывает значение  
ширины полигонов в  
соответствующую  
колонку атрибутов

**Функция *CalcWidth*:**  
рассчитывает  
максимальную /  
минимальную ширину  
данного полигона

**Функция *intersecLength*:**  
возвращает длину  
пересечения полигона и  
сканирующей линии в  
данной точке

**Функция *mergeLines*:**  
объединяет линии,  
полученные в  
результате пересечения  
сканирующей линии  
полигоном, если эти  
линии имеют общие  
точки

**Функция *point2tuple*:**  
конвертирует объекты  
класса *QgsPoint* в  
кортежи типа (x, y)



# Реализация в Python + QGIS API

**Функция *main*:**  
проверка параметров  
введённых пользователем;  
инициализация функции  
***azimuthWidth*** с заданными  
параметрами

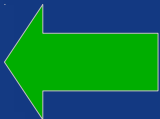
**Функция *azimuthWidth*:**  
записывает значение  
ширины полигонов в  
соответствующую  
колонку атрибутов

**Функция *CalcWidth*:**  
рассчитывает  
максимальную /  
минимальную ширину  
данного полигона

**Функция *intersecLength*:**  
возвращает длину  
пересечения полигона и  
сканирующей линии в  
данной точке

**Функция *mergeLines*:**  
объединяет линии,  
полученные в  
результате пересечения  
сканирующей линии  
полигоном, если эти  
линии имеют общие  
точки


**Функция *point2tuple*:**  
конвертирует объекты  
класса *QgsPoint* в  
кортежи типа (x, y)



# Пример выполнения

```
:~> python ./width.py poly.shp width 285.8 max abs Mix 1.5
```

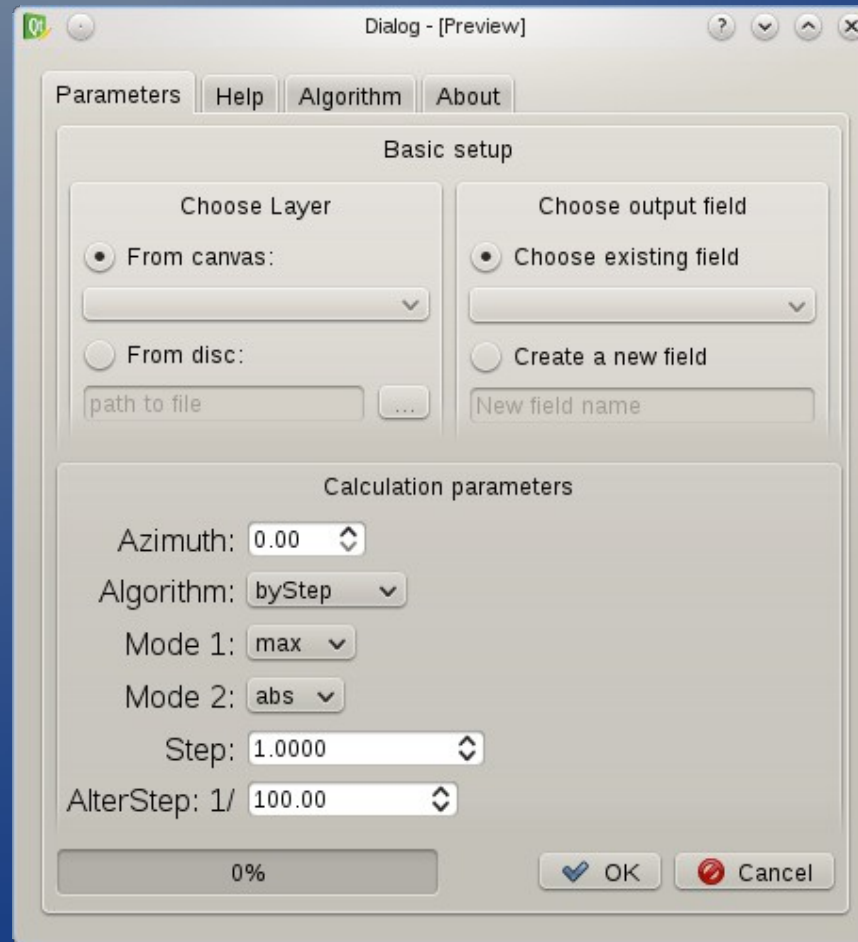
```
layer loaded!  
column found  
column creation skipped  
geometry ID: 0 | max width: 8200.98271785  
geometry ID: 1 | max width: 7527.12924347  
geometry ID: 2 | max width: 2113.4327195  
geometry ID: 3 | max width: 0  
geometry ID: 4 | max width: 67.7008817268  
geometry ID: 5 | max width: 69.6009238843  
geometry ID: 6 | max width: 66.4389510205
```



Невалидная  
геометрия

# Планы на будущее

- Внедрение многопоточности.
- Создание полноценного модуля QGIS:



# Лицензия

GPL 3

# Спасибо за внимание!

Скачать утилиту можно здесь:

<http://ssrebelious.blogspot.com/2012/09/azimuth-width-script.html>

Рябов Ю.В. <riabovvv@gmail.com >