

АСОИ поиска алгоритмов распознавания изоморфизма графов с помощью генетического программирования

Актуальность темы: задача проверки отношения изоморфизма двух графов является одной из двух нерешенных вычислительных задач, для которых не найдены алгоритмы решения за полиминальное время и которые не являются при этом NP-полными задачами.

Нахождение полиминального алгоритма значительно ускорит решение таких практических задач как:

- Проверка отношения эквивалентности знаний, хранимых в семантических сетях.
- Проверка соответствия электрических схем заданному шаблону
- Оптимизация программ при компиляции и оптимизация планов выполнения SQL запросов

Перечень задач, решенных в процессе проектирования:

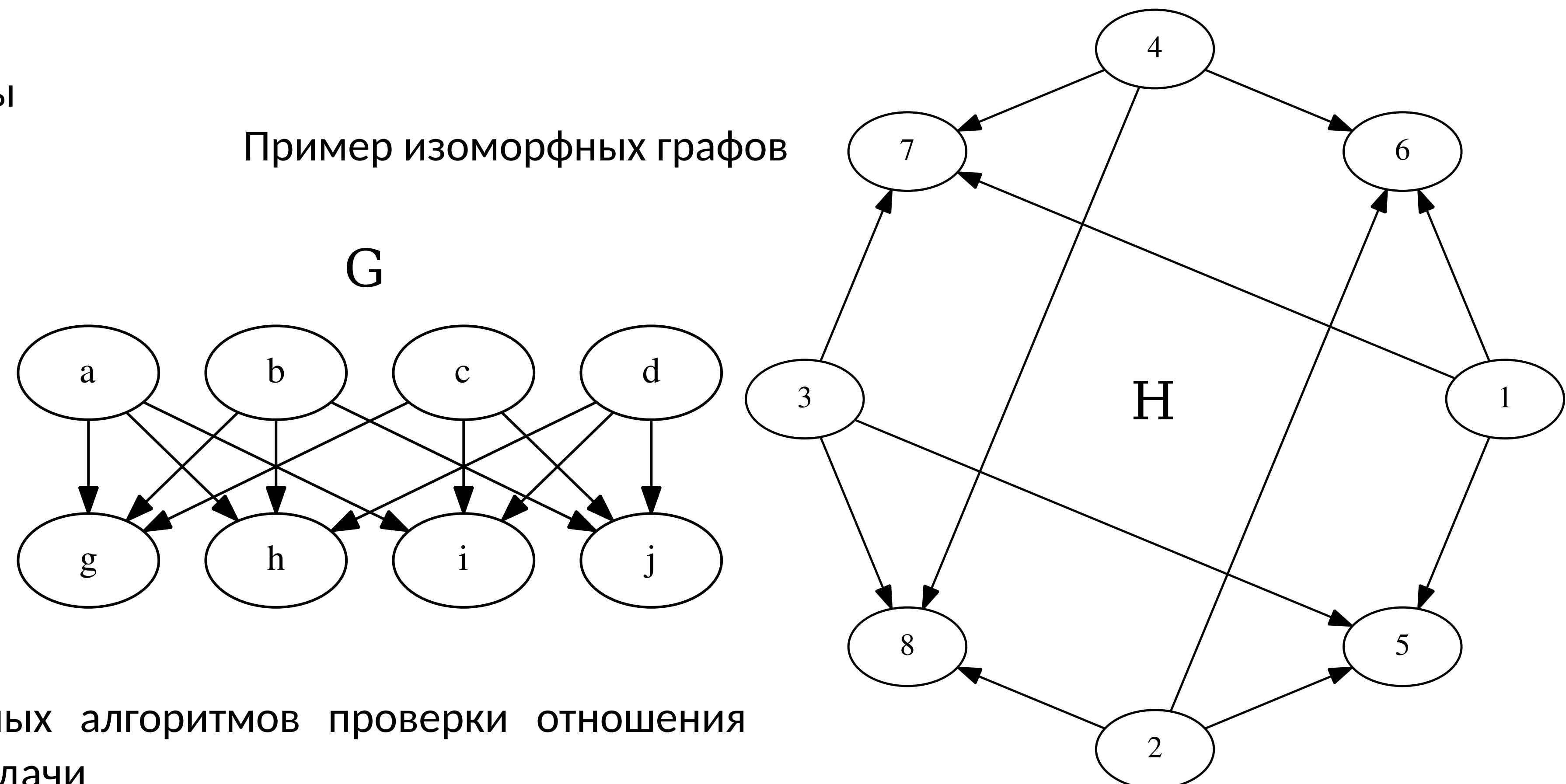
- Исследование предметной области проектирования
- Определение функциональных задач
- Изучение метода Генетическое программирование
- Разработка проблемно-ориентированного языка для внутреннего представления алгоритмов
- Выбор и обоснование критериев качеств оценки работы АСОИ и найденных алгоритмов
- Разработка схемы данных
- Разработка алгоритмов программы
- Разработка программы
- Отладка программы
- Разработка графического интерфейса пользователя
- Тестирование программы
- Разработка конструкторской и эксплуатационной документации

Цель: Осуществлять автоматический поиск эффективных алгоритмов проверки отношения изоморфизма среди всех возможных решений данной задачи

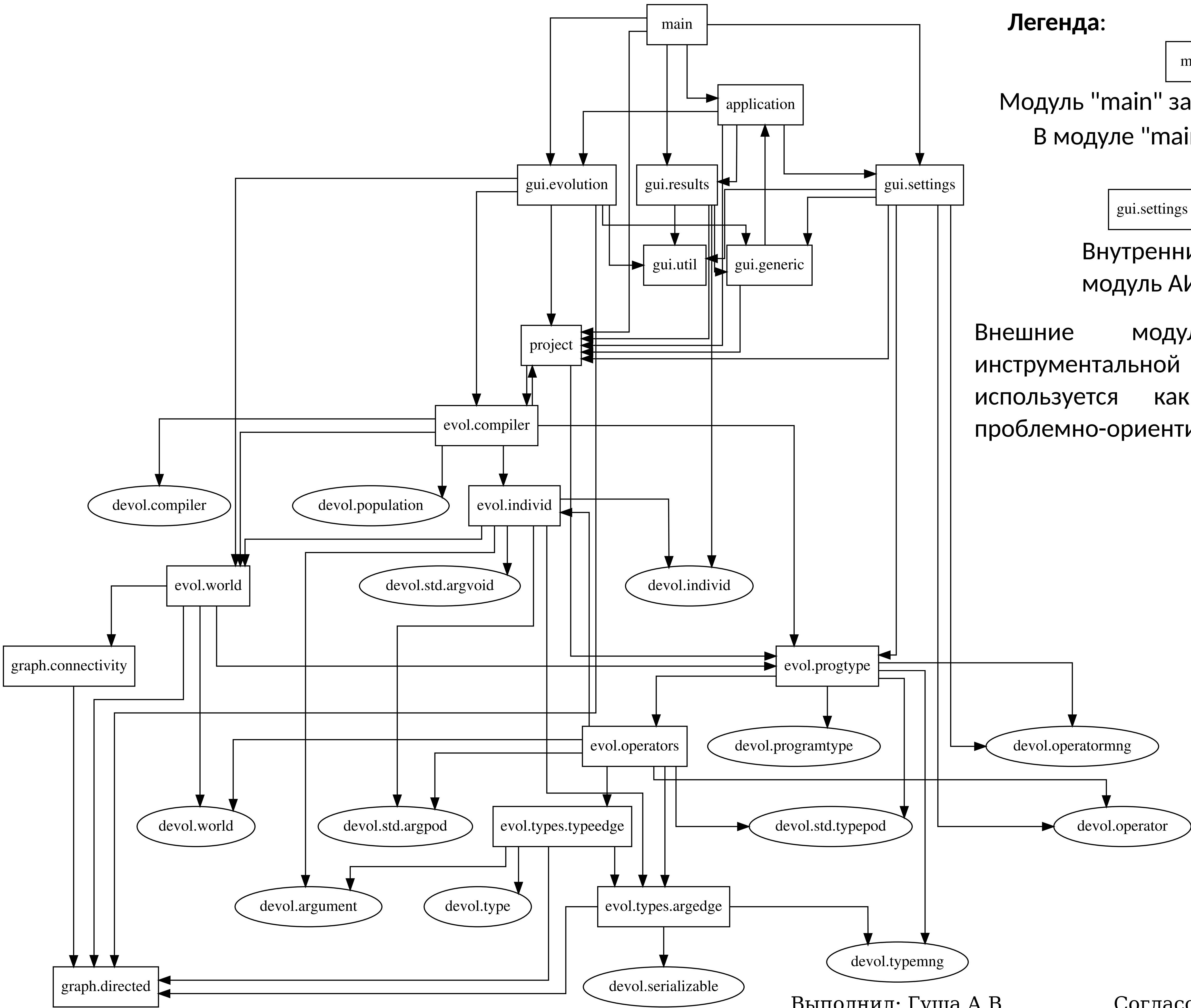
Изоморфизм - биекция между вершина особого вида

$$f : G \leftrightarrow H \quad \begin{pmatrix} a & b & c & d & g & h & i & j \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}$$

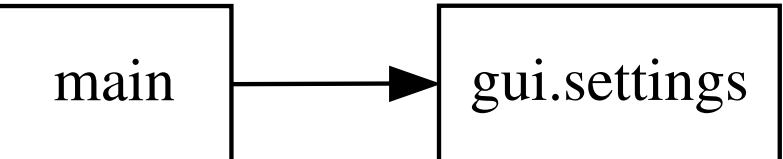
Пример изоморфных графов



Структура модулей АСОИ



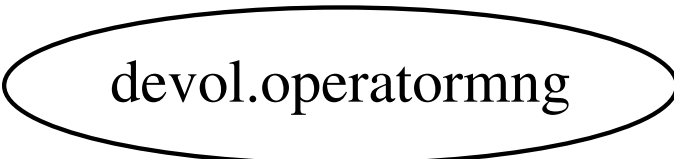
Легенда:



Модуль "main" зависит от модуля "gui.settings"
В модуле "main" есть "import gui.settings"



Внутренний
модуль АИС

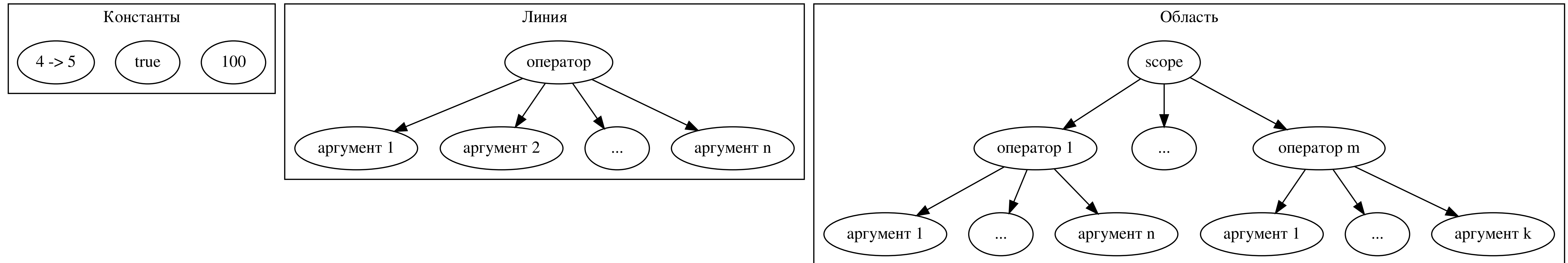


Внешний
модуль АИС

Внешние модули являются модулями
инструментальной библиотеки Devol, которая
используется как основа для реализации
проблемно-ориентированного языка.

Описание проблемно-ориентированного языка генетического программирования

Программы строятся из операторов, каждый из которых имеет несколько слотов-аргументов определенного типа. Каждый аргумент может быть константой, линией (line) или областью (scope). Линии в области вычисляются последовательно.



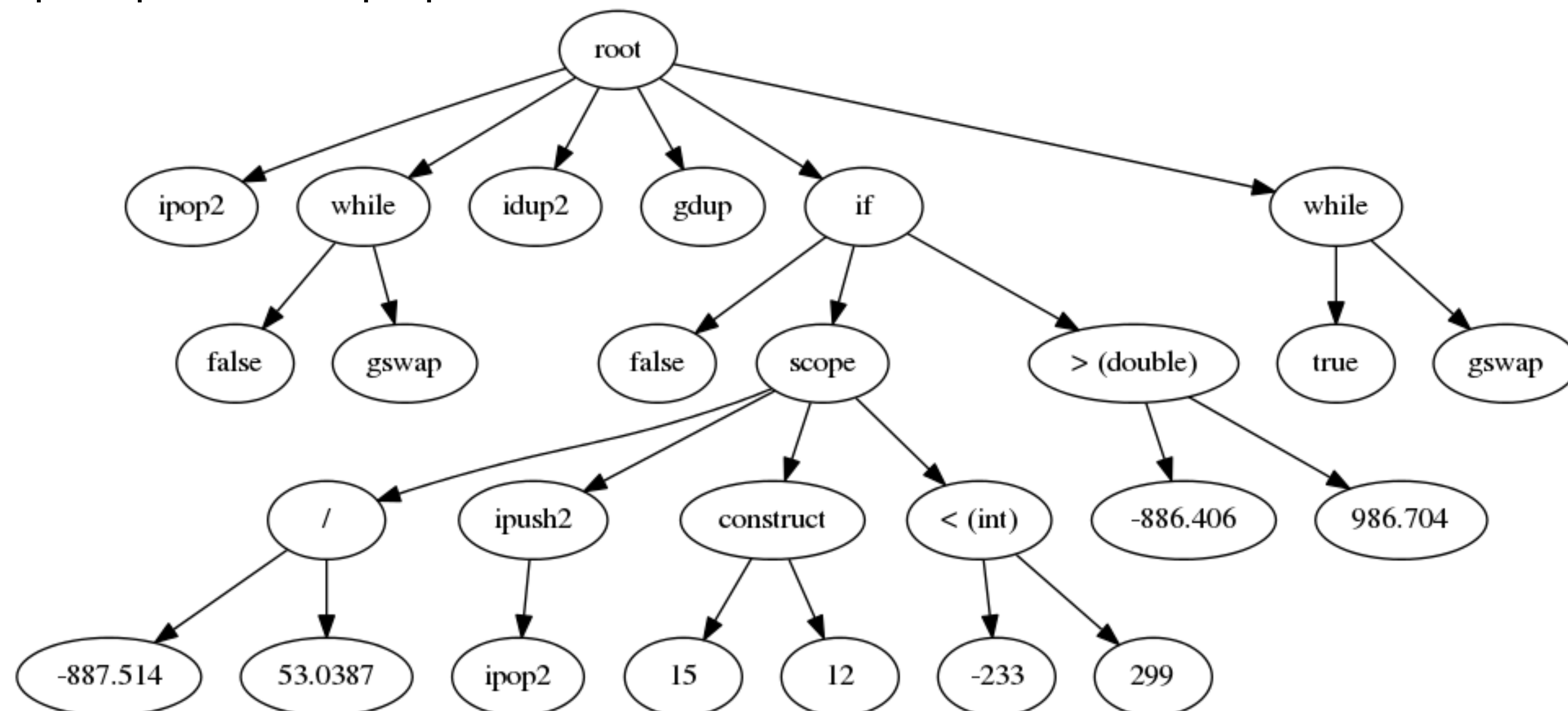
Данный язык является императивным с побочными эффектами, интерпретируется во время выполнения программы. Язык имеет строгую статическую типизацию, что необходимо для генерации изначально корректных программ.

Каждый оператор считается оператором с побочными эффектами

В данном языке определены следующие типы:

- Целочисленные числа (integer)
- Действительные числа (double)
- Логические значения (boolean)
- Пустой тип (void)
- Ребро графа вида integer -> integer

Пример типовой программы:



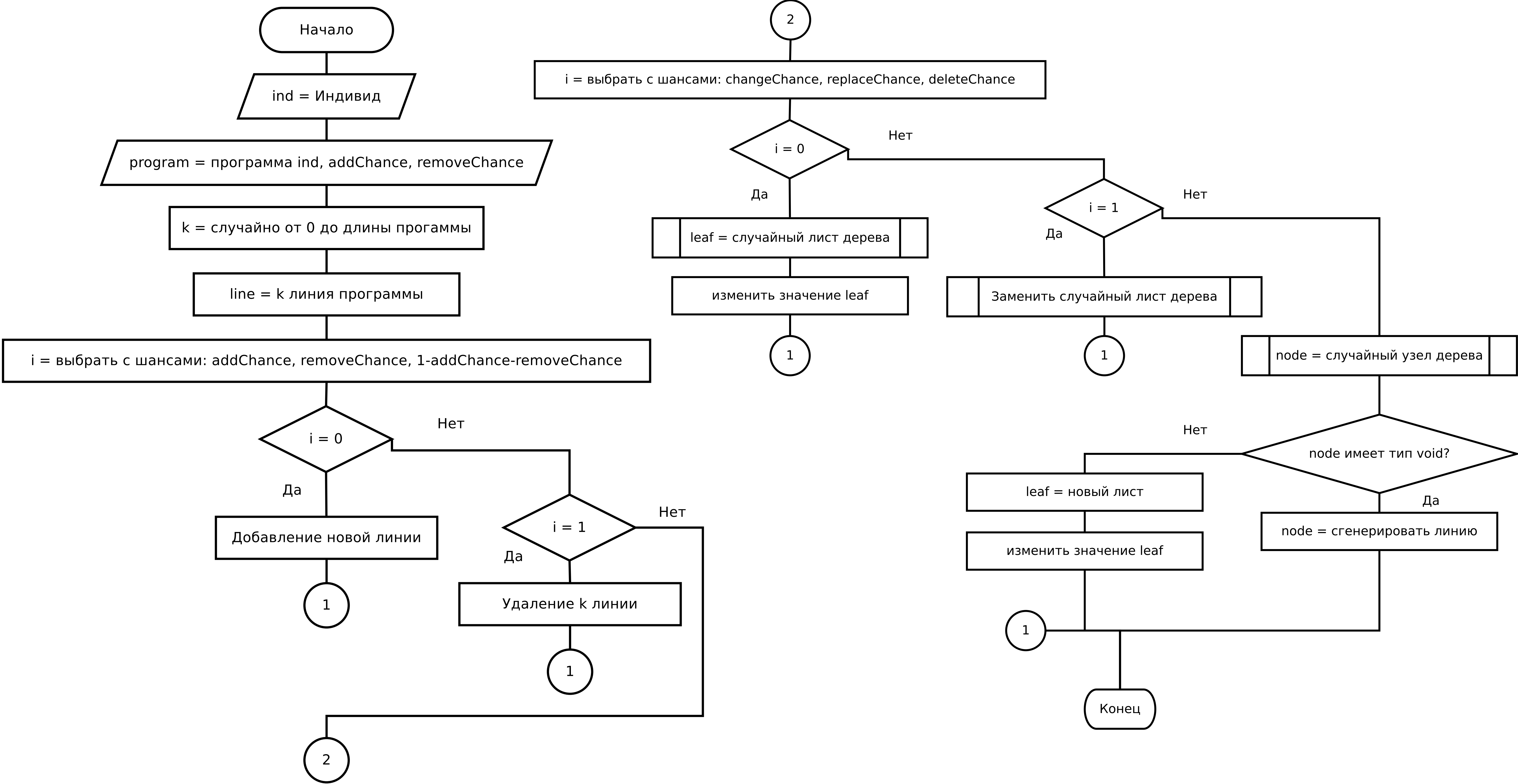
Программам доступны следующие операции:

- Условный оператор (if)
- Оператор цикла (while)
- Операторы работы со стеком общего назначения
- Операторы для работы с двумя стеками, хранящими входные графы
- Арифметические операции (+, *, /)
- Логические операции (&&, ||, !)
- Операции преобразования типов
- Операции работы с ребрами графов
- Операции для записи ответа

Выполнил: Гуца А.В. _____ Согласовано: Филиппович Ю.Н. _____

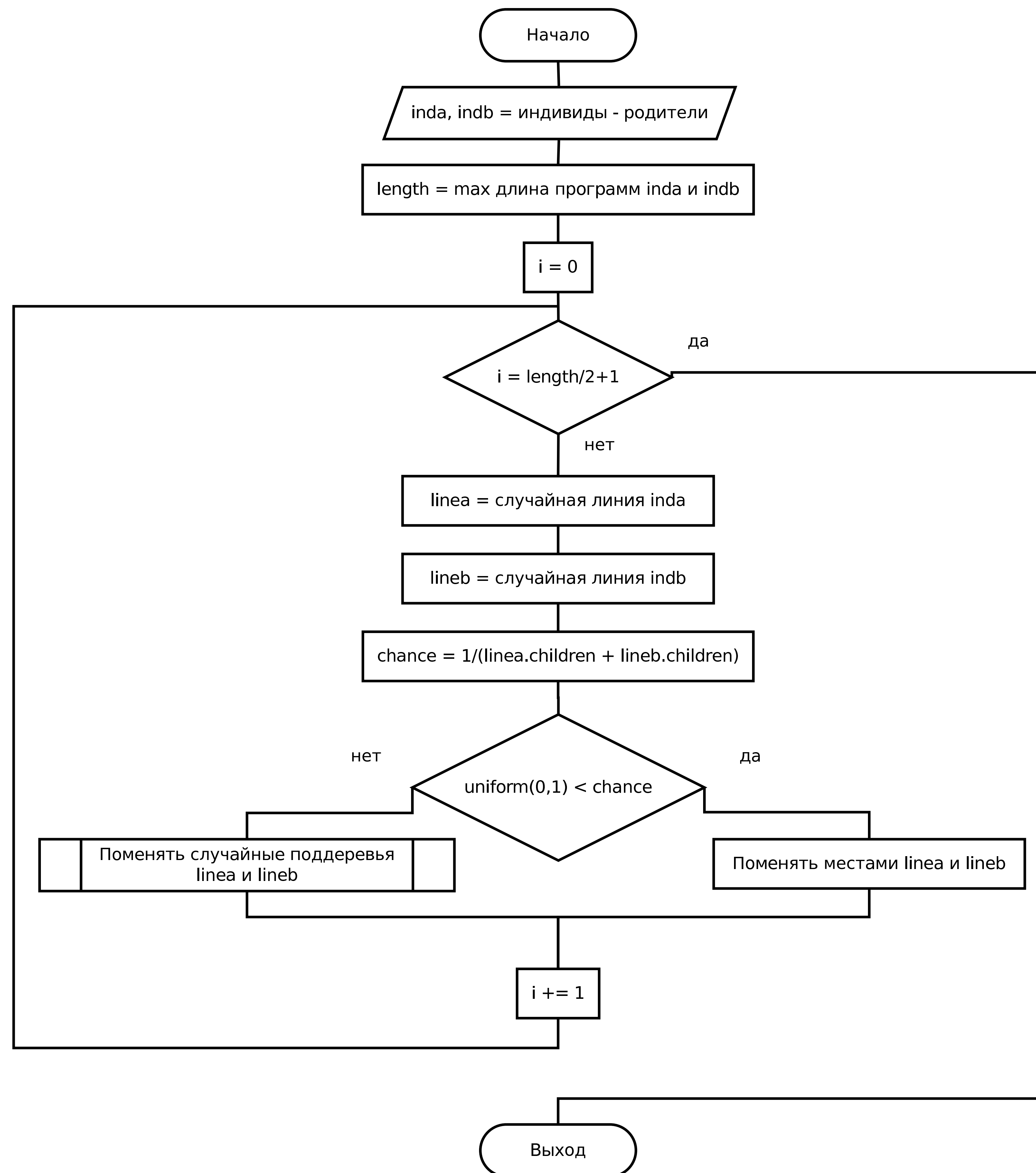
Основные алгоритмы АСОИ

Мутация



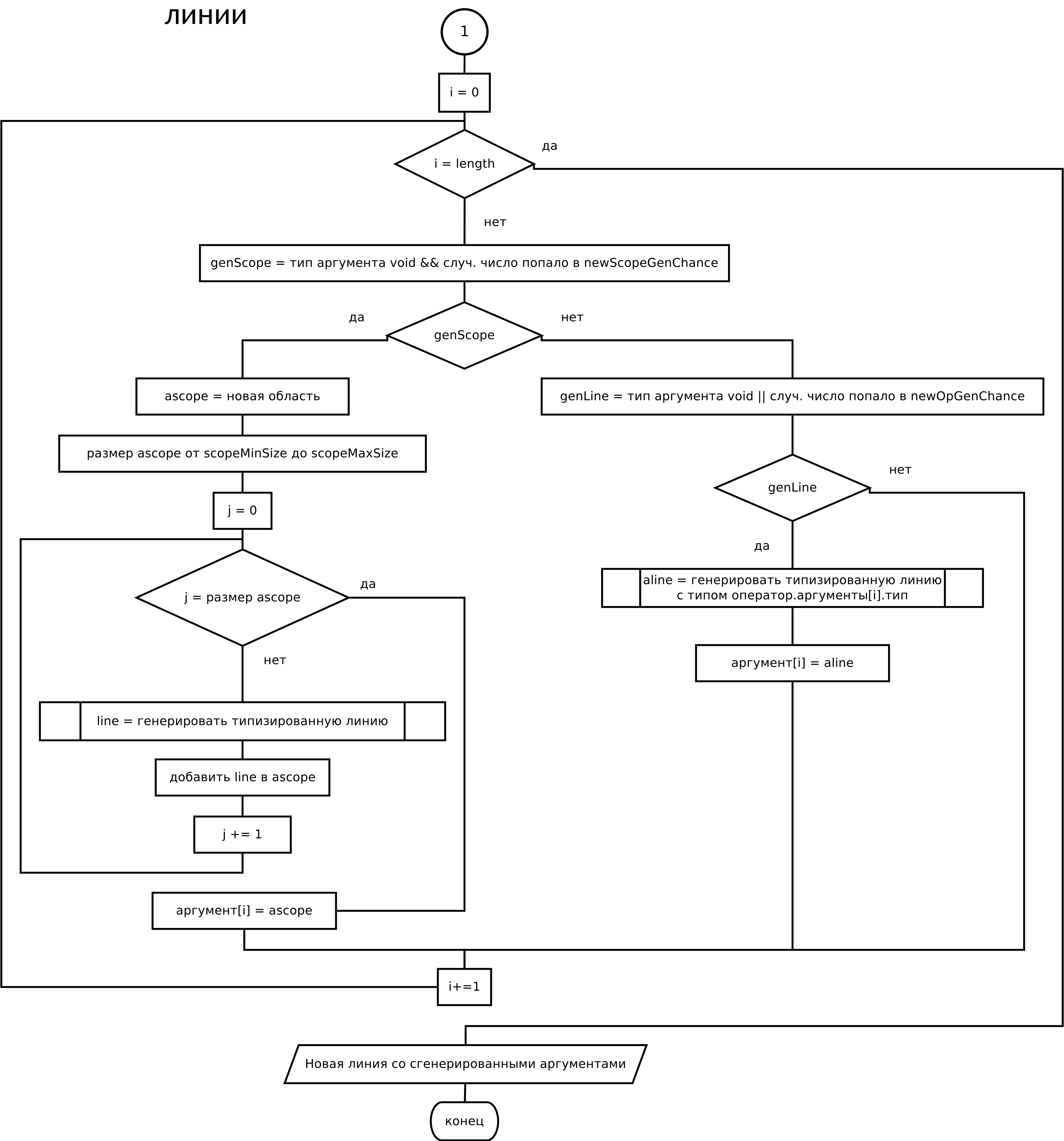
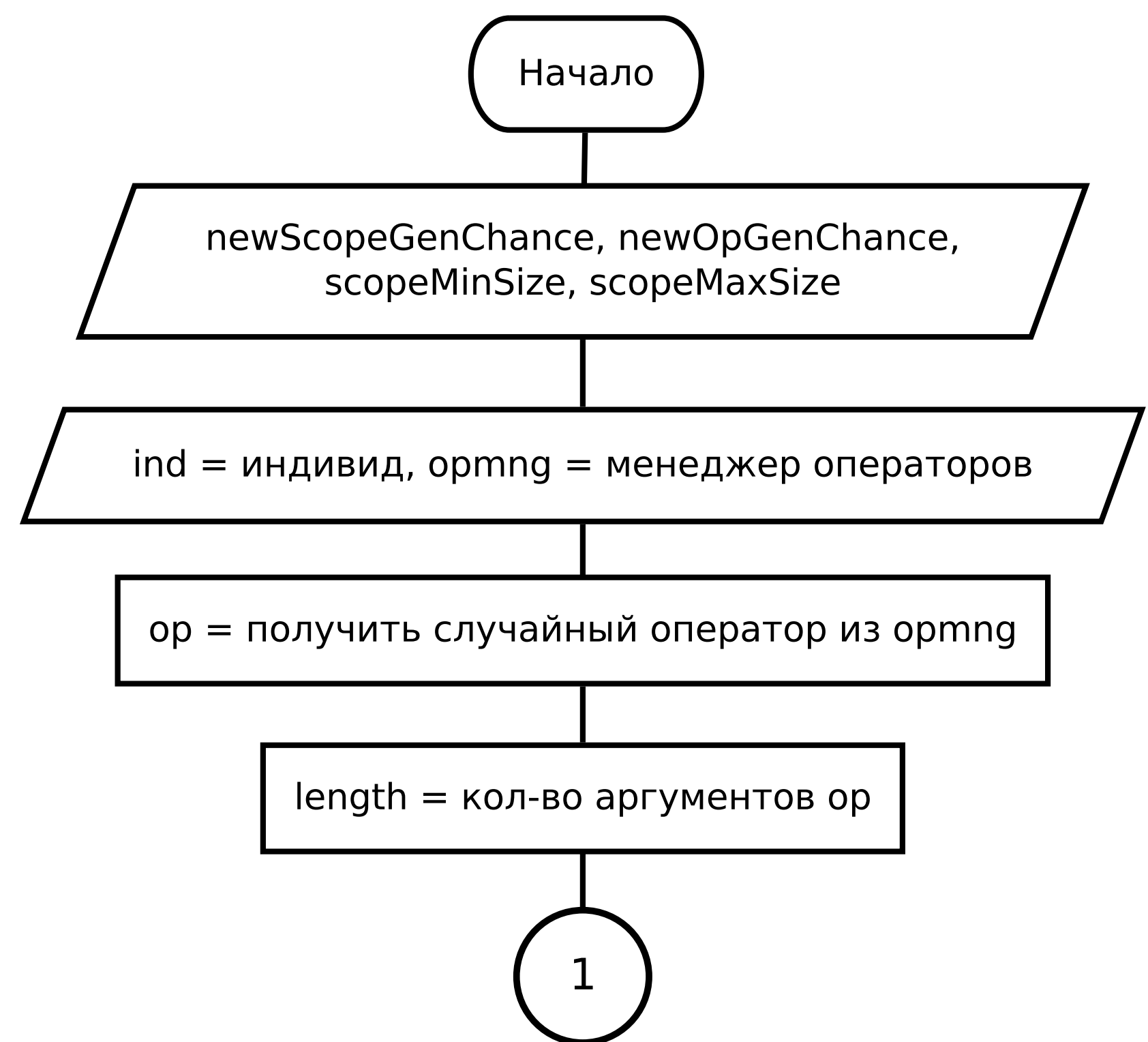
Основные алгоритмы АСОИ

Кроссинговер



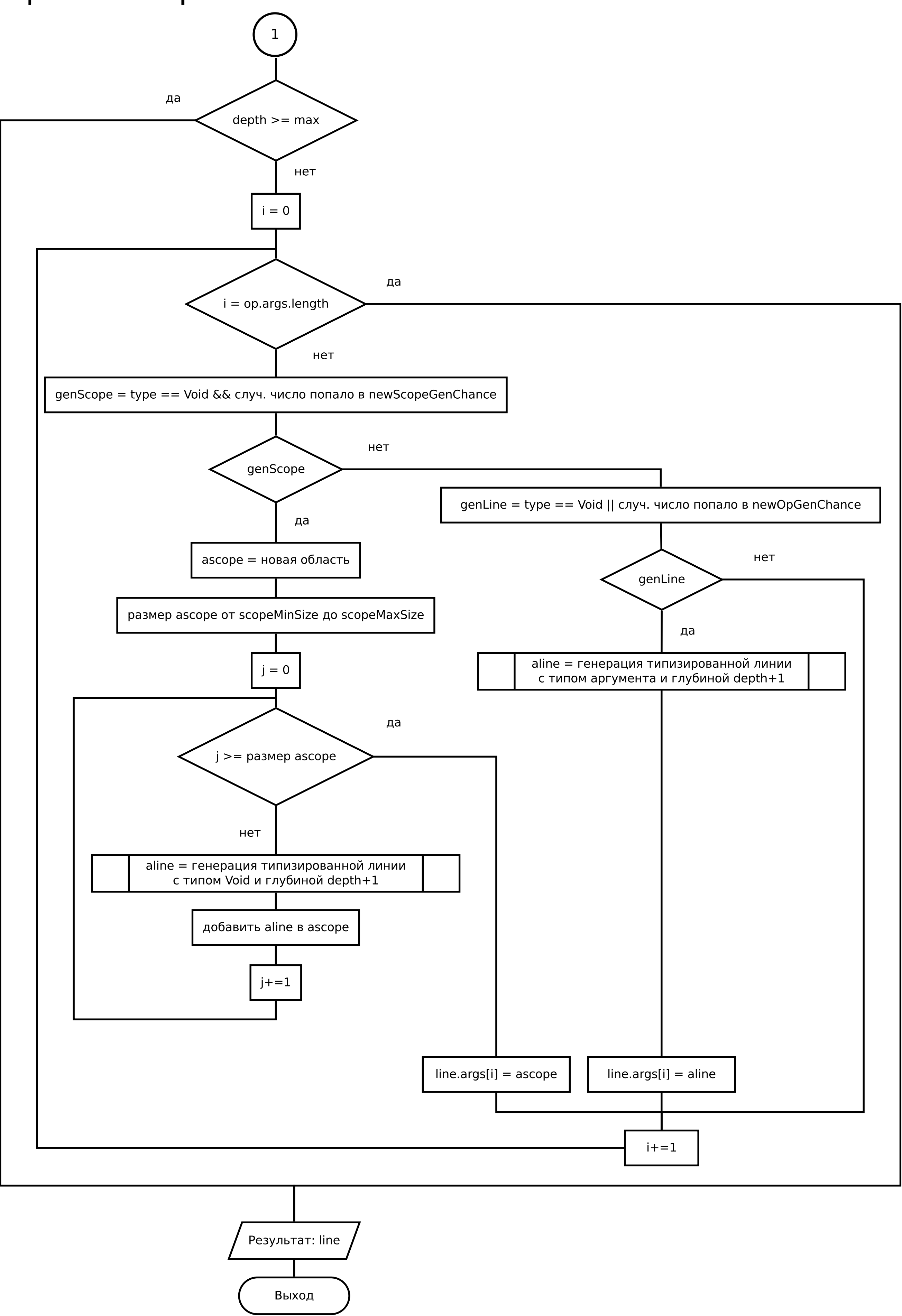
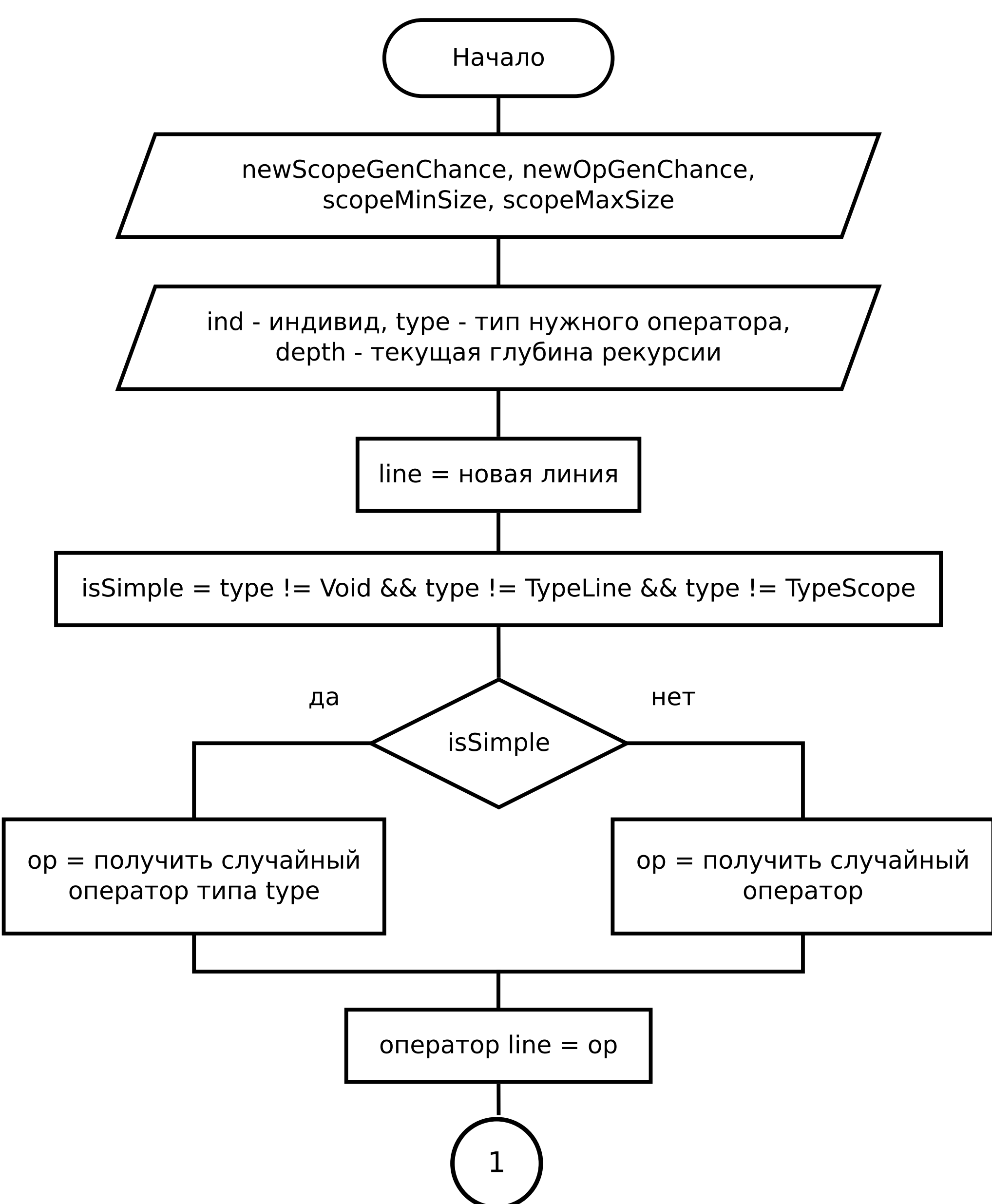
Основные алгоритмы АСОИ

Генерация нетипизированной линии



Основные алгоритмы АСОИ

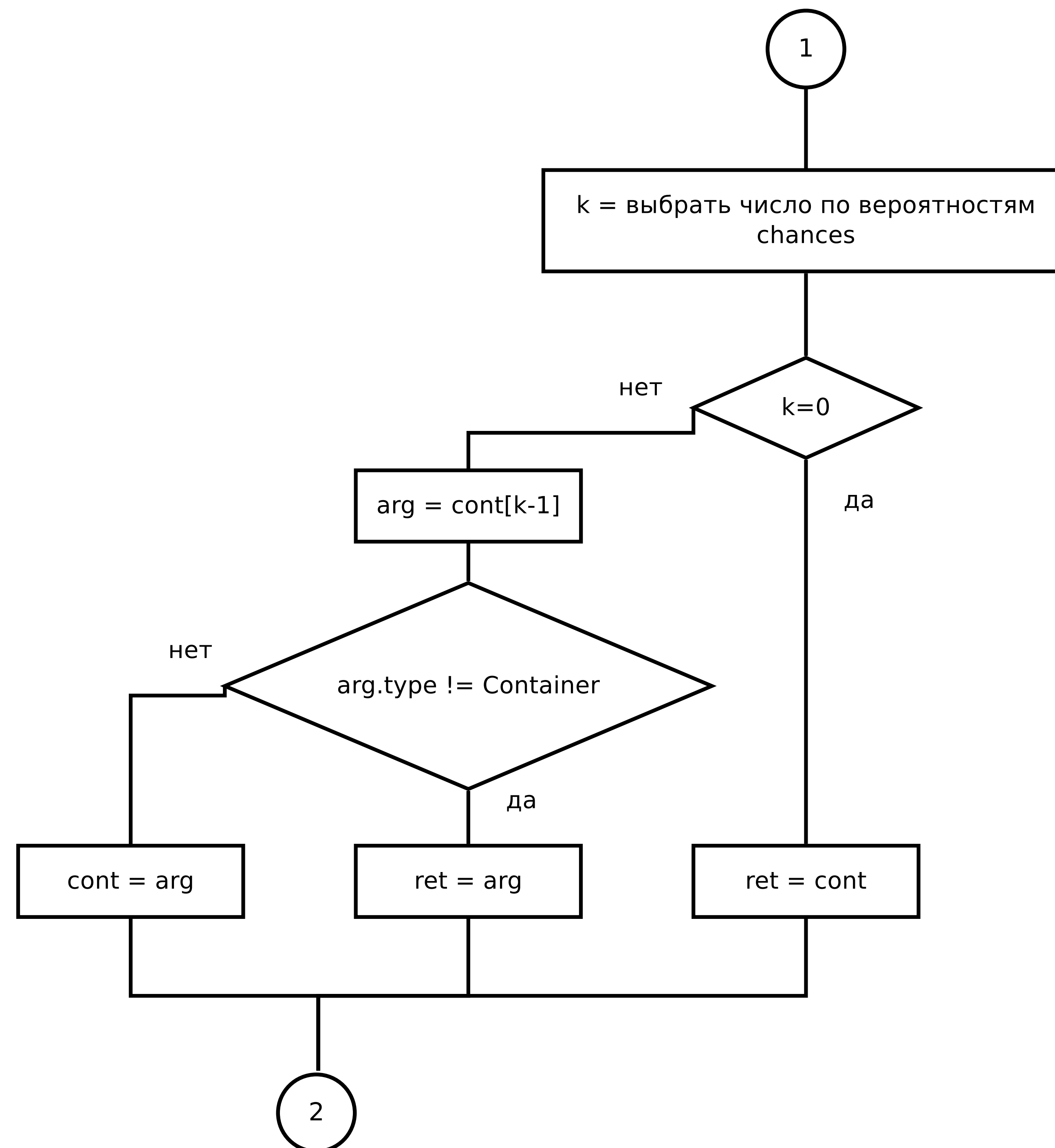
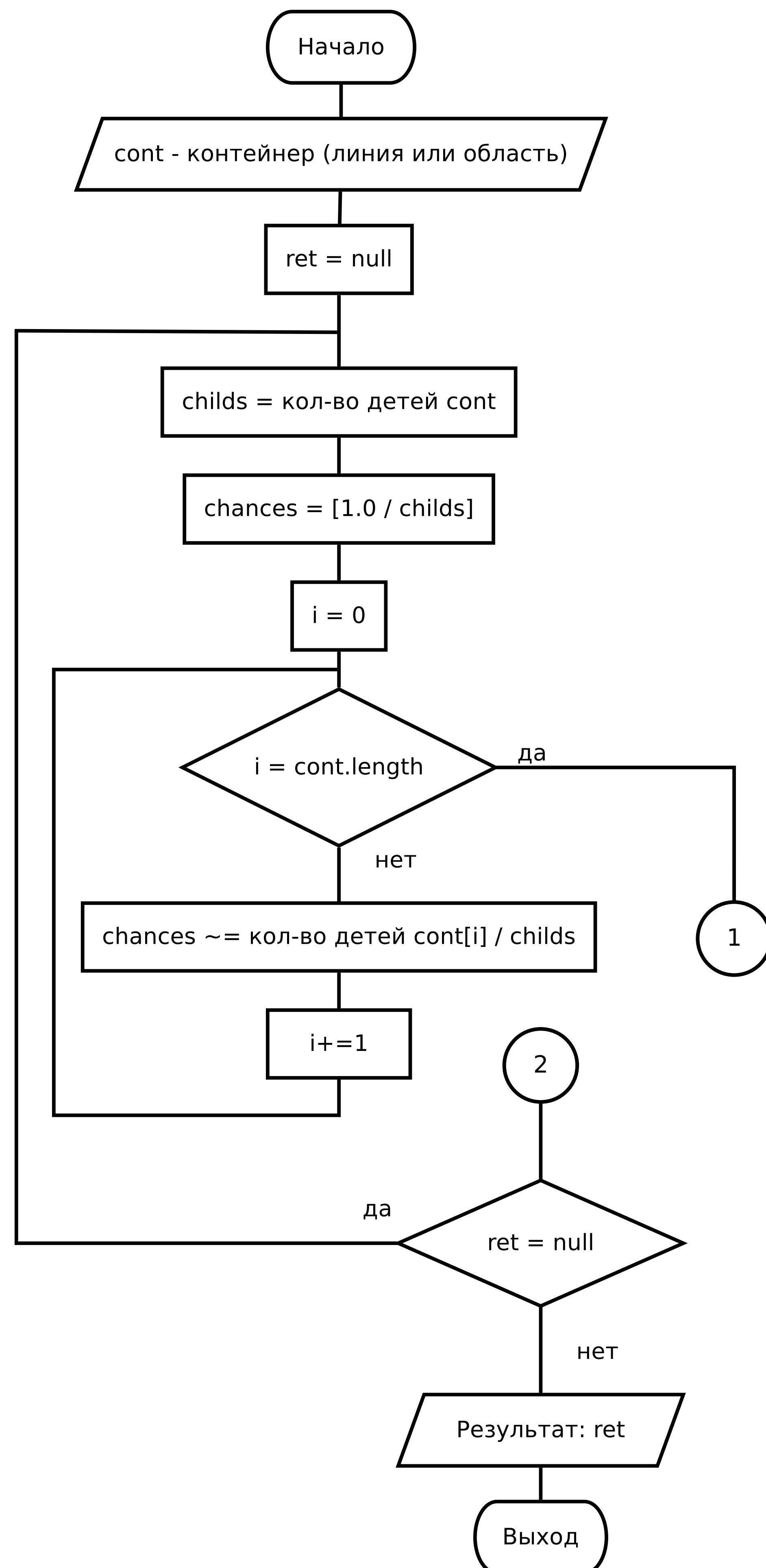
Генерация типизированной линии



Выполнил: Гуца А.В. _____ Согласовано: Филиппович Ю.Н. _____

Основные алгоритмы АСОИ

Выбор случайного узла дерева



Основные алгоритмы АСОИ

Формирование новой популяции

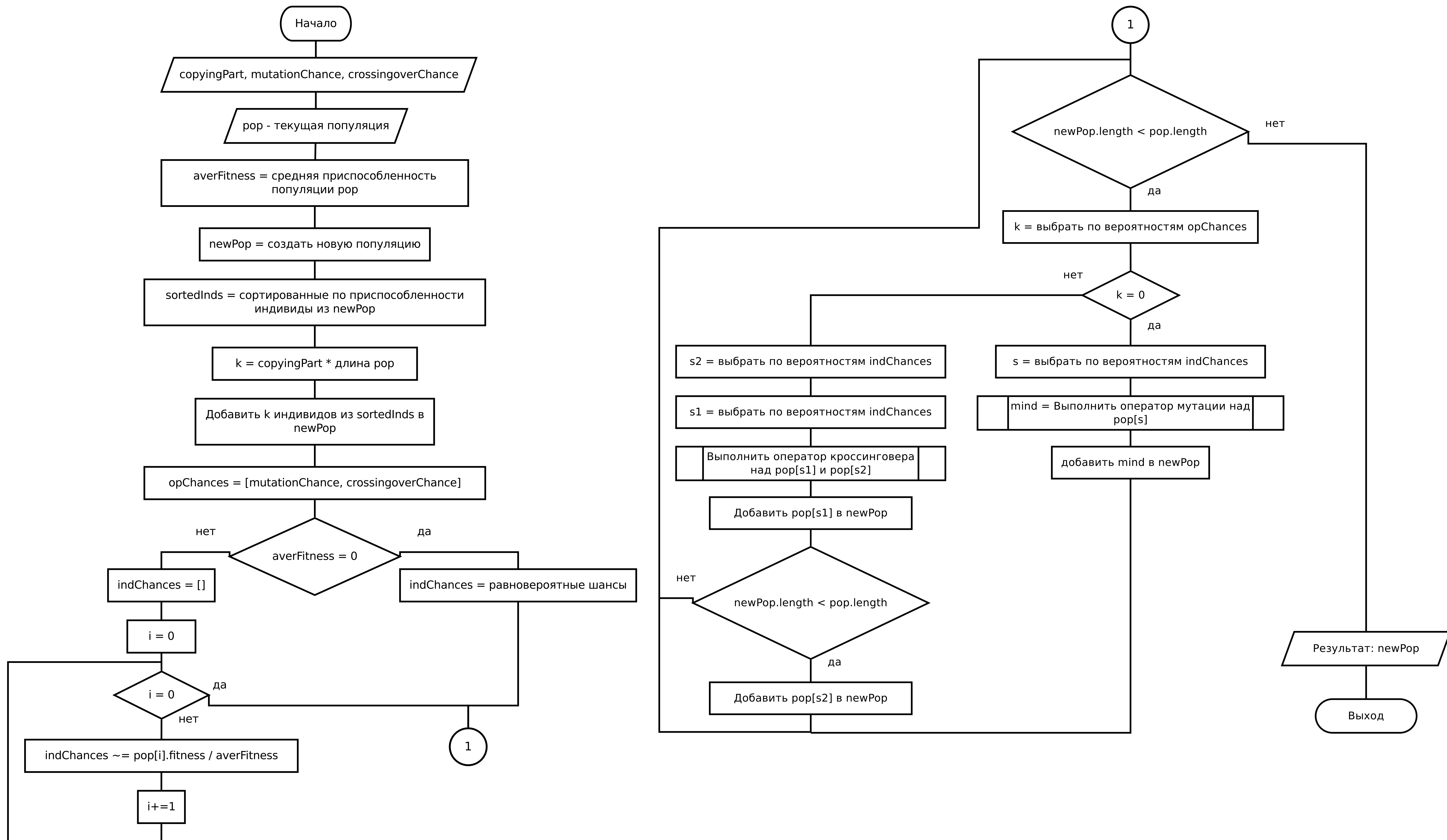
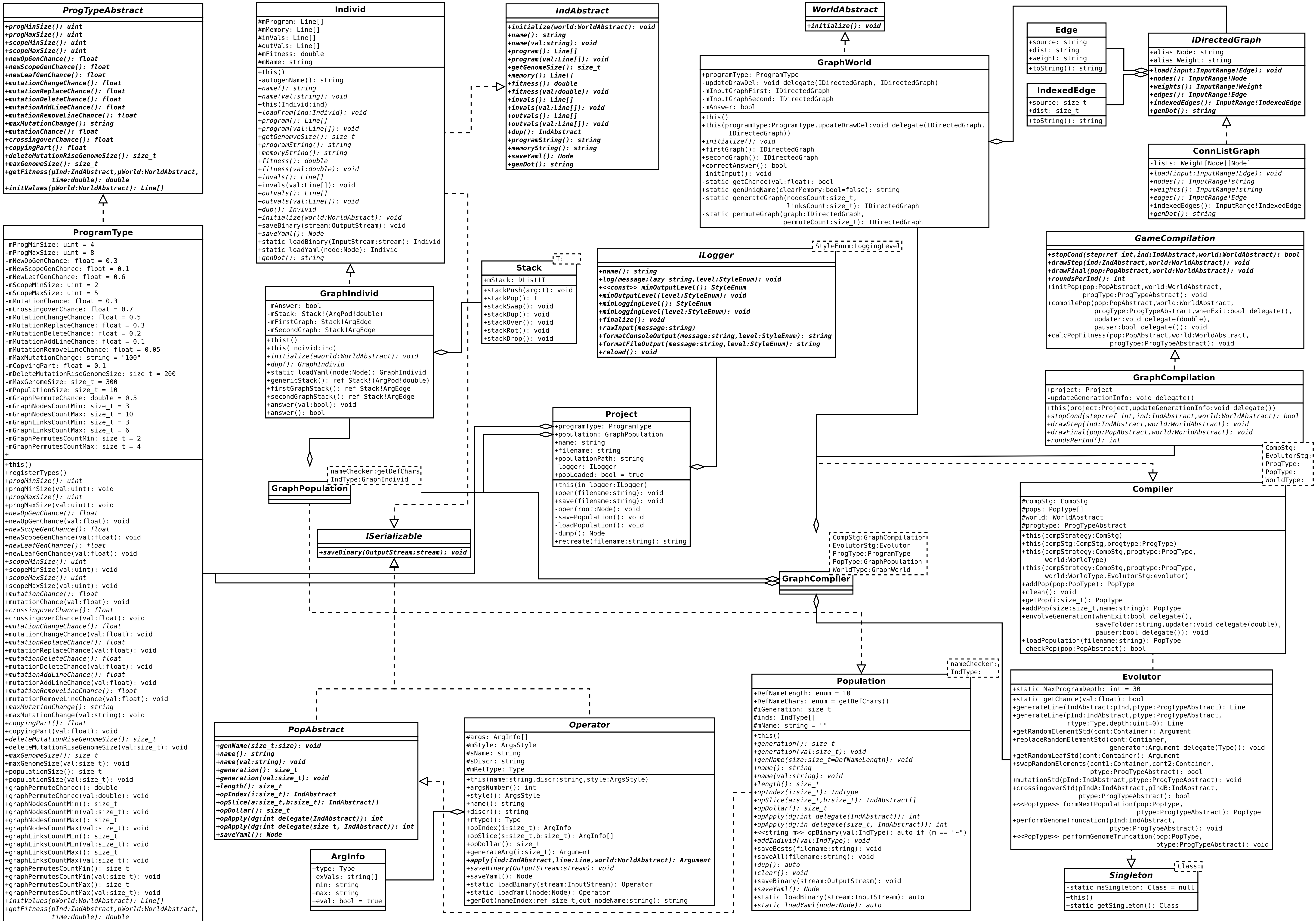


Диаграмма основных классов



Граф диалога с пользователем

