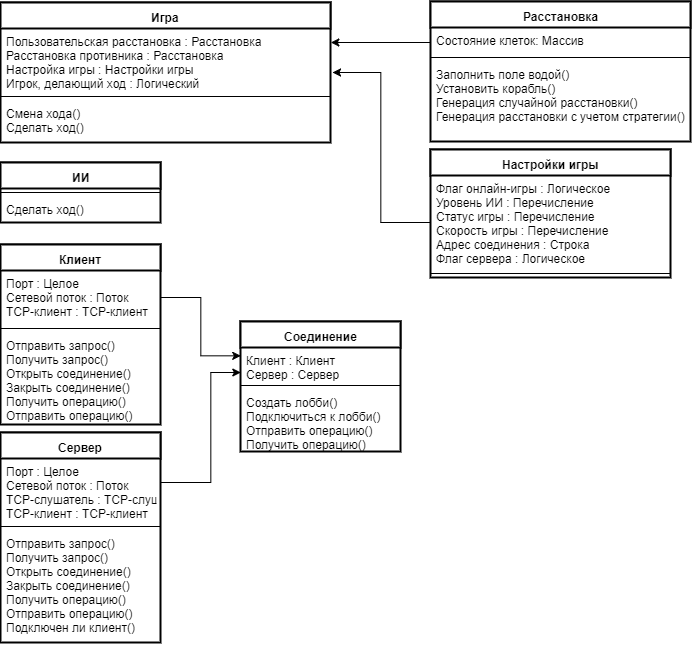
1. Разработка структур данных и классов

Диаграмма классов служит для представления статической структурной модели системы в терминологии классов объектно-ориентированного проектирования. Диаграмма классов может отражать различные взаимосвязи между отдельными сущностями предметной области, а также описывает их внутреннюю структуру и типы отношений. При этом на данной диаграмме не указывается информация о временных аспектах системах.

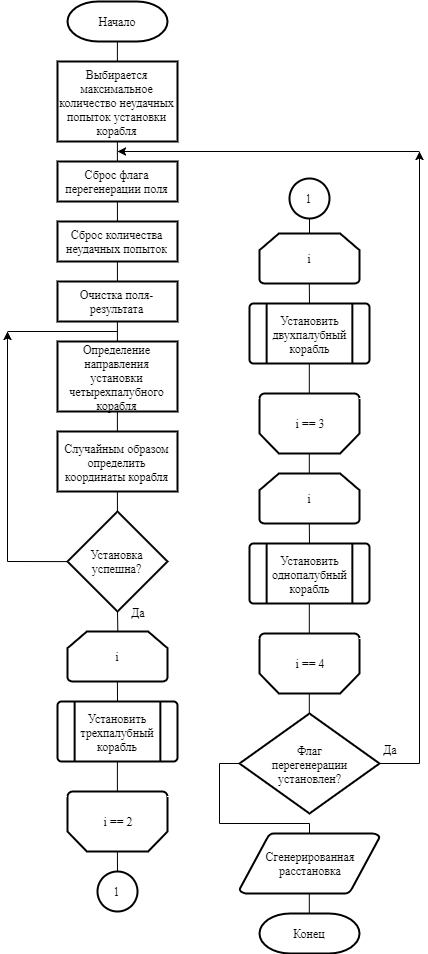
Класс в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов. Описание класса состоит в определении атрибутов и методов [14].

  
Рисунок 27 – Диаграмма классов системы

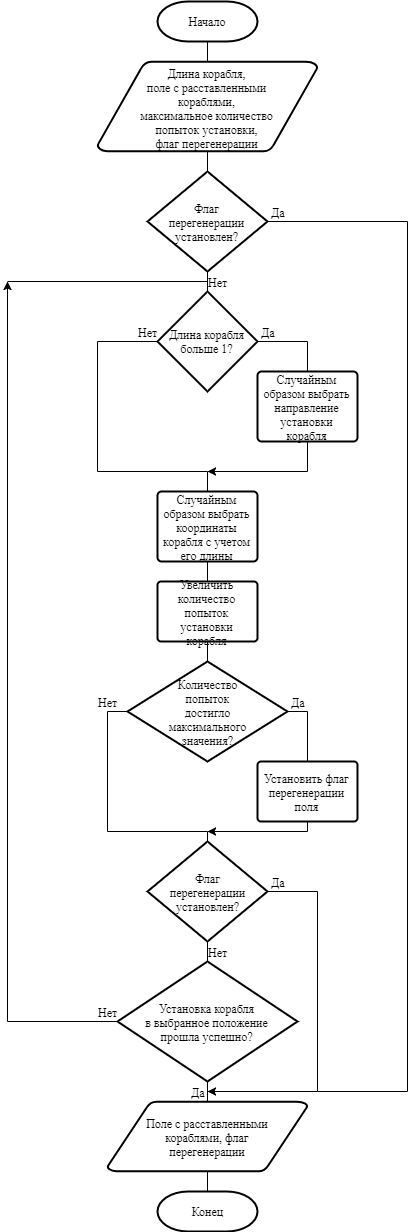
1. Выбор и обоснование алгоритмов обработки данных

Алгоритм – это конечный набор предписаний для получения решения задачи посредством конечного количества операций [16]. Многие задачи могут быть решены с использованием различных путей; в таком случае наиболее подходящее решение выбирается исходя из различных критериев.

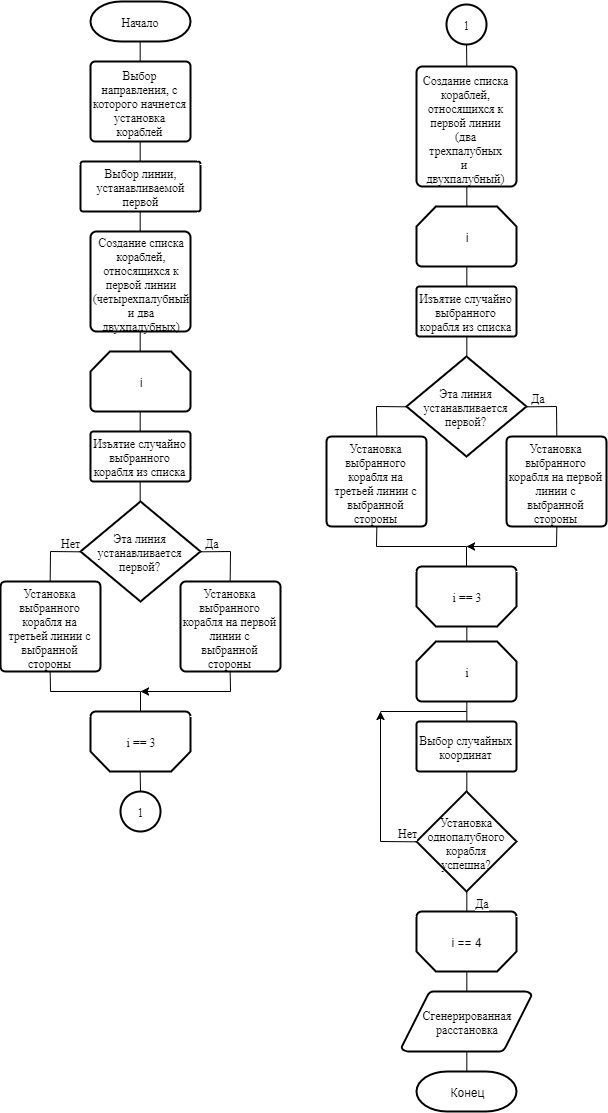
На рисунке 28 представлен алгоритм расстановки кораблей на поле случайным образом.

  
Рисунок 28 – Схема алгоритма расстановки кораблей случайным образом

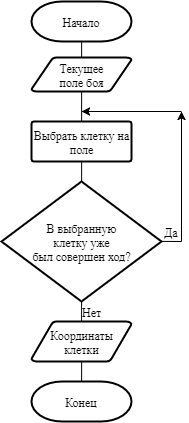
Для упрощения алгоритма в отдельную подфункцию выделена установка отдельного корабля. Алгоритм установки корабля по случайным координатам представлен на рисунке 29.

  
Рисунок 29 – Схема алгоритма установки корабля по случайным координатам

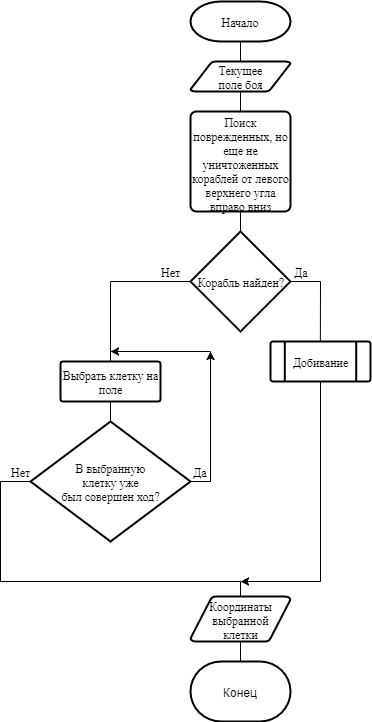
Помимо расстановки кораблей на поле случайным образом будет существовать возможность расстановки коралей в соответствии со стратегией, разработанной Я.И. Перельманом. Алгоритм генерации такой расстановки приведен на рисунке 30.

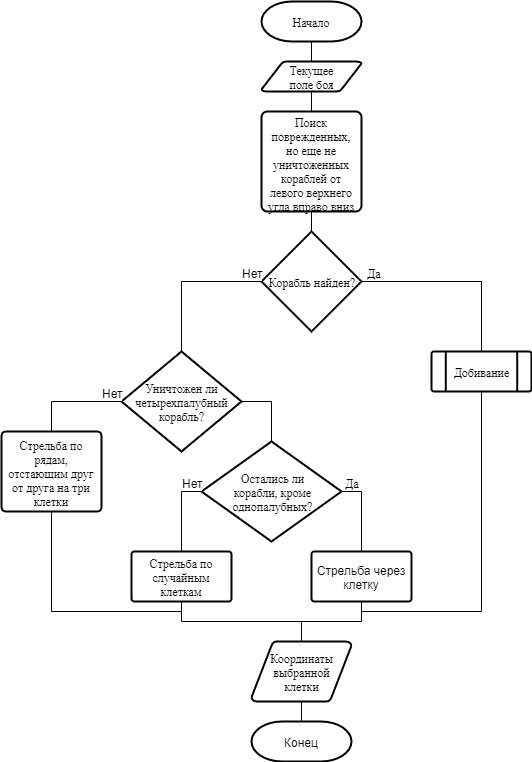
  
Рисунок 30 – Схема алгоритма расстановки кораблей в соответствии со стратегией.

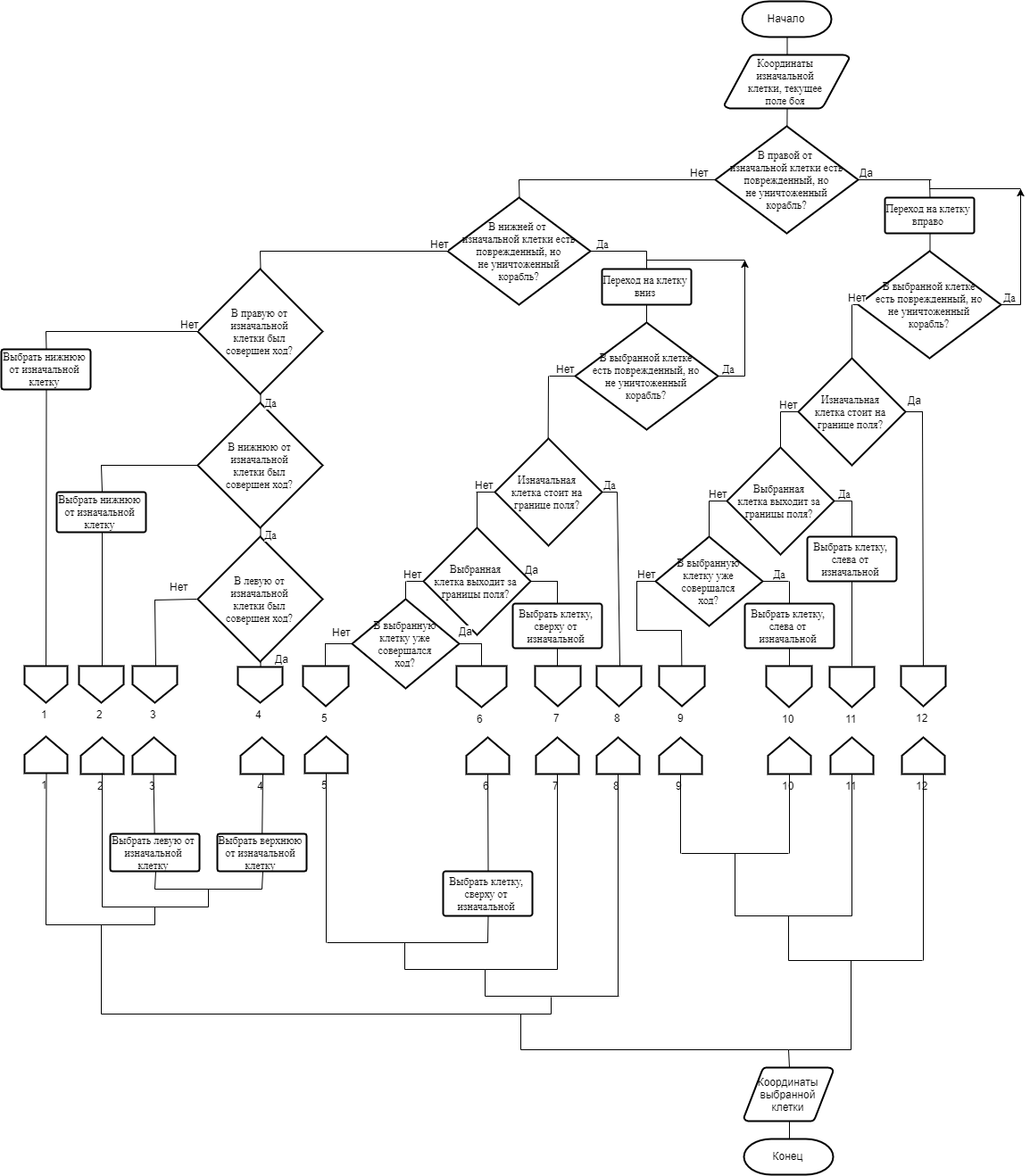
Подсистема ИИ должна содержать несколько реалиаций, соответствующих трем различным уровням сложности. Алгоритмы совершения хода ИИ легкой, средней и высокой сложности представлены соответственно на рисунках 31, 32, 33. Алгоритмы ИИ средней и высокой сложности используют подпрограмму добивания кораблей, алгоритм которой приведен на рисунках 34, 35.

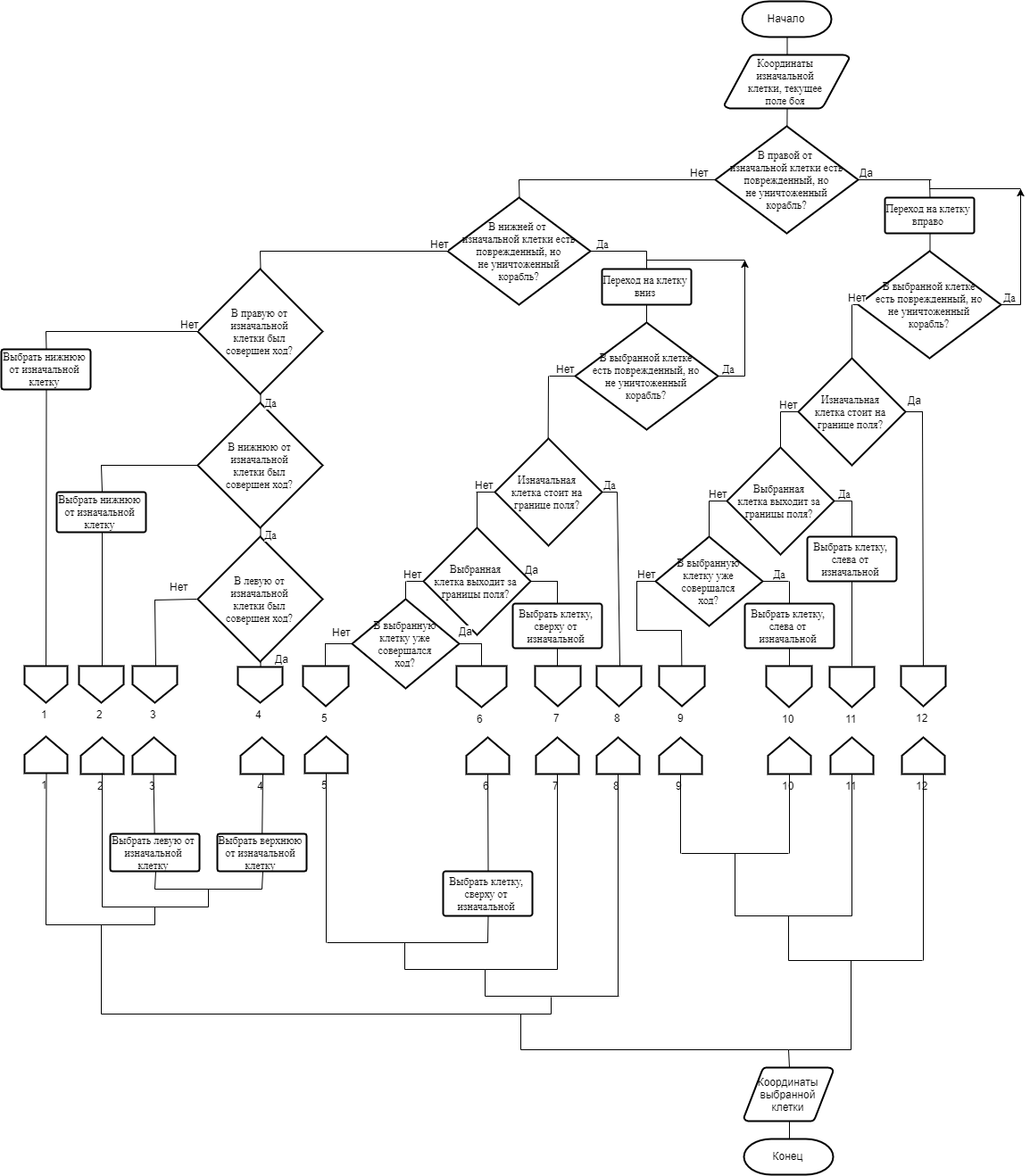
  
Рисунок 31 – Схема алгоритма совершения хода ИИ легкой сложности

текст

  
Рисунок 32 – Схема алгоритма совершения хода ИИ средней сложности

  
Рисунок 33 – Схема алгоритма совершения хода ИИ высокой сложности

  
Рисунок 34 – Схема алгоритма добивания корабля

  
Рисунок 35 – Продолжение схемы алгоритма добивания корабля