**Класс Cell  
Назначение:**

Cell — базовая ячейка игрового поля в сапёре. Она представляет собой не мину и может содержать информацию о количестве соседних мин. Класс управляет состоянием клетки: открыта ли она, помечена ли флагом и сколько мин рядом.

**Поля:**

* isOpen — открыта ли ячейка.
* isFlagged — стоит ли на ячейке флаг.
* mineCount — количество мин вокруг этой ячейки.

**Методы:**

* Конструктор инициализирует закрытую ячейку без флага и с нулём мин вокруг.
* open() — открывает ячейку.
* toggleFlag() — ставит или убирает флаг.
* isMine() — проверяет, является ли ячейка миной (в данной реализации — всегда false, так как это базовый класс).
* getMines() — возвращает количество мин вокруг ячейки.
* getIsOpen() — проверяет, открыта ли ячейка.
* getIsFlagged() — проверяет, стоит ли флаг.
* setMines(count) — устанавливает число соседних мин (приватный, доступен только для класса Field).

**Связь:**

* класс Field имеет доступ к приватным членам Cell, включая setMines(), isOpen, isFlagged, и mineCount.
* Используется в классе Field для хранения и управления состоянием клеток на игровом поле.

**Класс Bomb**  
**Назначение:**  
Bomb — специализированная ячейка игрового поля в сапёре, представляющая собой мину. Наследуется от класса Cell и переопределяет поведение для обозначения, что клетка — мина.

**Поля:**  
Наследует все поля базового класса Cell:

* isOpen — открыта ли ячейка.
* isFlagged — стоит ли на ячейке флаг.
* mineCount — количество мин вокруг этой ячейки (используется для обычных ячеек, для мин — неактуально).

**Методы:**

* Конструктор Bomb() — вызывает конструктор базового класса Cell, инициализирует клетку как закрытую и без флага.
* isMine() — переопределённый метод, возвращает true, что обозначает эту ячейку как мину.

**Связь с другими сущностями:**

* Наследуется от класса Cell, расширяя его функционал.
* Используется в классе Field для создания мин при генерации игрового поля.
* В логике игры проверяется через метод isMine() для определения поражения при открытии клетки.

**Класс Field**  
**Назначение:**  
Field — класс, реализующий игровое поле для сапёра. Управляет сеткой ячеек (Cell и Bomb), их состояниями, генерацией мин, подсчётом соседних мин, логикой открытия клеток, установки флагов, а также отслеживанием состояния игры (выигрыш, проигрыш).

**Поля:**

* rows — количество строк игрового поля.
* cols — количество столбцов игрового поля.
* mineCount — количество мин на поле.
* openedCells — количество уже открытых ячеек.
* field — двумерный вектор указателей на объекты Cell (или наследников, например Bomb).
* firstMove — флаг, указывающий, был ли сделан первый ход (чтобы избежать установки мин в стартовой области).
* won — флаг победы.
* gameOver — флаг окончания игры (проигрыш или победа).

**Методы:**

* Конструктор Field(rows, cols, mineCount) — создаёт поле заданного размера, инициализирует ячейки объектами Cell, устанавливает начальные состояния.
* Деструктор ~Field() — освобождает память, удаляя все объекты ячеек.
* generateMines(safeX, safeY) — рандомно размещает мины, избегая области вокруг координат safeX, safeY (защита первого хода).
* calculateNeighbourMines() — подсчитывает количество мин вокруг каждой клетки, не являющейся миной, и устанавливает это число через setMines().
* inBounds(x, y) — проверяет, находятся ли координаты внутри границ поля.
* openSafeCells(x, y) — рекурсивно открывает соседние клетки без мин, если их количество соседних мин равно нулю.
* openCell(x, y) — открывает клетку, если это первый ход — генерирует мины и считает соседей, проверяет, не мина ли это, и вызывает открытие соседних безопасных клеток; обновляет состояние игры.
* toggleFlag(x, y) — ставит или снимает флаг на клетке, если она не открыта.
* isWon() — возвращает, выиграна ли игра.
* isGameOver() — возвращает, завершена ли игра.
* display() — выводит игровое поле в консоль с цветовой подсветкой, показывая открытые клетки, мины, флаги и закрытые клетки.

**Связь с другими сущностями:**

* Хранит объекты классов Cell и Bomb через указатели, управляет их состояниями.
* Использует методы классов Cell и Bomb для открытия, флагов, определения мин.
* Обеспечивает игровую логику, взаимодействует с классами Cell и Bomb для корректного отображения и обработки игрового процесса.

**Класс GameSettings**  
**Назначение:**  
GameSettings — класс, отвечающий за хранение и инициализацию настроек игры: размеров игрового поля (строк и столбцов) и количества мин. Обрабатывает аргументы командной строки для установки пользовательских параметров.

**Поля:**

* rows — количество строк игрового поля.
* cols — количество столбцов игрового поля.
* mines — количество мин на поле.

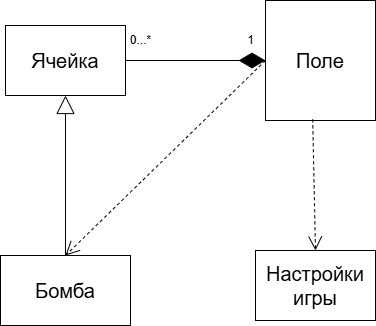
**Методы:**

* Конструктор GameSettings() — задаёт значения по умолчанию: 9 строк, 9 столбцов, 10 мин.
* parseArguments(argc, argv) — парсит аргументы командной строки:
  + argv[1] — задаёт количество строк (если есть).
  + argv[2] — задаёт количество столбцов (если есть).
  + argv[3] — задаёт количество мин (если есть).
  + Ограничивает максимальное количество мин до половины от общего числа клеток, выводит предупреждение, если передано слишком много мин.
  + Корректирует значения, если заданы некорректные (например, меньше 1), возвращая значения по умолчанию.

**Связь с другими сущностями:**

* Используется при создании игрового поля (Field) для задания параметров игры.
* Взаимодействует с внешним вводом (командной строкой), преобразуя строки в числовые настройки.

**Объектная модель**

****