Игра "Жизнь".

Игру «Жизнь» придумал американский математик Джон Хортон Конуэй (хотя его научные интересы вполне серьезны, например, теория групп и теория чисел). Возникающие в её процессе ситуации очень похожи на реальные процессы, происходящие при зарождении, развитии и гибели колонии живых организмов. По этой причине «Жизнь» можно отнести к быстро развивающейся в наши дни категории игр, имитирующих процессы, происходящие в живой природе.

Правила игры.

Действия игры происходят в размеченной на клетки и ограниченной плоскости. Каждая клетка на этой поверхности может находиться в двух состояниях: быть «живой» или быть «мёртвой» (пустой). Клетка имеет восемь соседей(окружающих клеток). Распределение живых клеток в начале игры называется первым поколением. Каждое следующее поколение рассчитывается на основе предыдущего по таким правилам:

в пустой (мёртвой) клетке, рядом с которой ровно три живые клетки, зарождается жизнь;

если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае (если соседей меньше двух или больше трёх) клетка умирает («от одиночества» или «от перенаселённости»).

Игра прекращается, если на поле не останется ни одной «живой» клетки, если при очередном шаге ни одна из клеток не меняет своего состояния (складывается стабильная конфигурация) или если конфигурация на очередном шаге в точности (без сдвигов и поворотов) повторит себя же на одном из более ранних шагов (складывается периодическая конфигурация).

Эти простые правила приводят к огромному разнообразию форм, которые могут возникнуть в игре.

Игрок не принимает прямого участия в игре, а лишь расставляет или генерирует начальную конфигурацию «живых» клеток, которые затем взаимодействуют согласно правилам уже без его участия (он является наблюдателем).

Реализация на компьютере.

Очевидно, для начала надо создать некий фундамент, то есть основу.

Фундаментом в данной программе является двумерный массив. Который играет роль размеченного и ограниченного пространства. Запустив среду программирования, я создал новый Java Project назвал его "Life". Далее я создал новый пакет и дал ему аналогичное название. В новом пакете я создал класс "OldGeneration" . Как видно из названия, этот класс будет содержать в себе "старое поколение". В этом классе есть четыре поля. Первое поле -это массив, который содержит в себе переменные Boolean, второе-количество строк (n), третье-количество столбцов (m), и наконец четвертое-количество живых клеток(l). У нас есть метод "input", в котором происходит заполнение массива(количество строк, столбцов и живых клеток мы выбираем сами). Причем есть два режима установки живых клеток: вручную и рандомно. Есть также метод "sysoPole", в котором создается новый массив с переменными типа String, старый массив копируется с заменой значения True на "L" и False на "\*" и выводится на экран. Это сделано для лучшей наглядности, так как старый массив с переменными типа Boolean неудобен для просмотра. Фундамент создан, теперь переходим к реализации правил игры, но перед этим нам надо создать два дополнительных массива. Для каждого создаем новый класс : "NewGeneration" и "ForAutostop". В классе "NewGeneration" будет создан новый массив такой же размерности и типом переменных как в классе "OldGeneration". Создание и заполнение массива осуществляет метод "additionalArray" . Первоначально все элементы этого массива равны False. В классе "ForAutostop" также создается новый массив такой же размерности и типом переменных как в классе "OldGeneration". Здесь метод "copy" копирует значения из массива класса "OldGeneration". То есть он будет запоминать предыдущее поколение. Все, теперь переходим к правилам. Я создал новый класс, который я назвал "Rules" . В нем был создан метод "check", в параметры передаются массивы из классов "OldGeneration" и "NewGeneration". В этом методе прописаны условия для проверки и подсчета живых клеток. Для крайних , угловых и центральных клеток прописаны отдельные условия. Массив начинает просматриваться с самого начала. Берутся элементы по порядку и для каждого подсчитывается количество живых "соседей". После подсчета проверяется условия: если мы подсчитали соседей живой клетки, то проверяем: если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае (если соседей меньше двух или больше трёх) клетка умирает («от одиночества» или «от перенаселённости»); если же мы подсчитали соседей мертвой клетки, то проверяем следующее условие: в пустой (мёртвой) клетке, рядом с которой ровно три живые клетки, зарождается жизнь.

Проверяется массив из класса "OldGeneration", а изменяется массив из "NewGeneration", поскольку рождение и гибель «организмов» происходит одновременно, новорожденные клетки никак не влияют на гибель и рождение остальных клеток, и поэтому, проверяя новую конфигурацию, необходимо уметь отличать их от клеток, перешедших из предыдущего поколения. После метода "check" идет метод "aNewGeneration". Здесь происходит смена поколения, то есть старое поколение обновляется. А если посмотреть на код, то значения из массива класса "NewGeneration" копируются в массив класса "OldGeneration" и количество живых клеток обновляется. И после копирования, массив класса "OldGeneration" пересоздается, то есть все его переменные обратно становятся False. Далее идет метод "autostop" , который проверяет совпадение предыдущего и настоящего поколения и принимает соответствующее значение True или False. После проверки выполняется метод " copy", который сохраняет настоящее(предыдущее) поколение. Правила реализованы ,теперь создаем класс "Play", который будет содержать метод "begin". Этот метод вызывает методы "check", " aNewGeneration", "SysoPole" И "autostop" пока не выполнится одно из следующих условий: если на поле не останется ни одной «живой» клетки, если при очередном шаге ни одна из клеток не меняет своего состояния (складывается стабильная конфигурация) или если конфигурация на очередном шаге в точности (без сдвигов и поворотов) повторит себя же на одном из более ранних шагов (складывается периодическая конфигурация). Если же выполнено одно из этих условий ,то игра прекращается. Для запуска был создан класс "Start". В котором объявляются объекты , запускаются методы заполнения массивов и собственно метод "Play" , который отвечает

за весь процесс игры. Итак алгоритм реализован, все работает. Для запуска игры запускаем класс "Start" и любуемся сменой поколений.