Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Кафедра «Программное обеспечение автоматизированных систем»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе**

по дисциплине «Объектно-ориентированный анализ и программирование»

на тему: «Проектирование и реализация программы с использованием объектно-ориентированного подхода»

(индивидуальное задание – вариант №07\_01)

Студент: Сальников Д.А.

Группа: ПрИн-367

Работа зачтена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «      » \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20      г.

Руководитель проекта, нормоконтроллер \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Литовкин Д.В.

Волгоград 2023 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Направление 09.03.04 «Программная инженерия»   
Кафедра «Программное обеспечение автоматизированных систем»

Дисциплина «Объектно-ориентированный анализ и программирование»

Утверждаю

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_ Орлова Ю.А.

**ЗАДАНИЕ**

**на курсовую работу**

Студент: Сальников Д.А.

Группа: ПрИн-367

1. Тема: «Проектирование и реализация программы с использованием объектно-ориентированного подхода» (индивидуальное задание – вариант №07\_01)

Утверждена приказом от «24» января 2021г. № 101-ст

2. Срок представления работы к защите « 01 »   июня  2023 г.

3. Содержание пояснительной записки:

формулировка задания, требования к программе, структура программы, типовые процессы в программе, человеко-машинное взаимодействие, код программы и модульных тестов

4. Перечень графического материала:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Дата выдачи задания «15» февраля 2023 г.

Руководитель проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Литовкин Д.В.

Задание принял к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сальников Д.А.

«15» февраля 2023 г.

**Содержание**

[1 Формулировка задания 4](#_Toc129776285)

[2 Нефункциональные требования 5](#_Toc129776286)

[3 Первая итерация разработки 6](#_Toc129776287)

[3.1 Формулировка упрощённого варианта задания 6](#_Toc129776288)

[3.2 Функциональные требования (сценарии) 7](#_Toc129776289)

[3.3 Словарь предметной области 11](#_Toc129776290)

[3.4 Структура программы на уровне классов 12](#_Toc129776291)

[3.5 Типовые процессы в программе 12](#_Toc129776292)

[3.6 Человеко-машинное взаимодействие 12](#_Toc129776293)

[3.7 Реализация ключевых классов 15](#_Toc129776294)

[4 Вторя итерация разработки 16](#_Toc129776295)

[4.1 Функциональные требования (сценарии) 16](#_Toc129776296)

[7.1.1. 4.2 Словарь предметной области 16](#_Toc129776297)

[4.3 Структура программы на уровне классов 16](#_Toc129776298)

[4.4 Типовые процессы в программе 16](#_Toc129776299)

[4.5 Человеко-машинное взаимодействие 16](#_Toc129776300)

[4.6 Реализация ключевых классов 21](#_Toc129776301)

[5 Список использованной литературы и других источников 21](#_Toc129776302)

# 1 Формулировка задания

Правила игры «Глупый робот и умный робот»:

* имеется лабиринт, в котором действует маленький умный робот и большой глупый робот, их позиции задаются;
* маленьким роботом управляет человек, а большим - компьютер;
* игра ведется пошагово;
* в лабиринте имеются болота, а также вертикальные и горизонтальные препятствия (на стыке клеток);
* цель маленького робота – достичь выхода, цель большого робота – поймать маленького робота;
* большой робот рационально действует только, когда маленький робот попадает в поле его действия размером 2×2, в остальных случаях робот идет сначала либо вправо, либо влево по направлению к маленькому роботу;
* маленькому роботу заходить в болото нельзя, а большой робот пропускает три хода.

Дополнительные требования:

* предусмотреть в программе **точки расширения**, используя которые можно реализовать вариативную часть программы (в дополнение к базовой функциональности).

Вариативность:

* предусмотреть возможность создания *стационарных / автономных*, *возобновляемых самостоятельно / возобновляемых извне / невозобновляемых* источников питания; а также возможность роботом использовать такие источники питания.

Реализовать:

* аккумулятор, у которого некоторое количество заряда теряется, если оно не использованное на очередном шаге;
* ветряная мельница, которая вырабатывает за один шаг некоторое количество заряда. Чтобы роботу зарядится от мельницы, он должен обладать автономным возобновляемым извне источником питания, находиться рядом с мельницей и выполнить команду подзарядки.

# 2 Нефункциональные требования

1. Программа должна быть реализована на языке Java SE 19 с использованием стандартных библиотек, в том числе, библиотеки Swing.
2. Форматирование исходного кода программы должно соответствовать Java Code Conventions, September 12, 1997.

# 3 Первая итерация разработки

### 3.1 Формулировка упрощённого варианта задания

Правила игры «Сломанный робот»:

* имеется лабиринт, в котором действует маленький умный робот и большой глупый робот, их позиции задаются;
* маленьким роботом управляет человек, а большим - компьютер;
* игра ведется пошагово;
* в лабиринте имеются болота, а также вертикальные и горизонтальные препятствия (на стыке клеток);
* цель маленького робота – достичь выхода, цель большого робота – поймать маленького робота;
* большой робот рационально действует только, когда маленький робот попадает в поле его действия размером 2×2, в остальных случаях робот идет сначала либо вправо, либо влево по направлению к маленькому роботу;
* маленькому роботу заходить в болото нельзя, а большой робот пропускает три хода.

### 3.2 Функциональные требования (сценарии)

*кому/что поручить*

**1) Сценарий** «Играть»

1. **По указанию пользователя** Игра стартует.
2. **По указанию** Игры Поле создаёт МхN поле. Поле создает Лабиринт.
3. **По указанию** Игры Лабиринт создаёт и размещает на Поле Стены, Маленького Робота, управляемого человеком, и Большого Робота, управляемого компьютером, Точку Выхода, размещает Болота.
4. **В ответ на запрос** Игры Поле **сообщает** о Маленьком и Большом Роботах, которые находятся на нём.
5. **Делать**
   1. **По указанию** **пользователя** Маленький Робот перемещается на соседнюю Ячейку.
   2. **Если** Маленький Робот перемещается на Точку Выхода, то она сообщает об этом Игре, что приводит к исчезновению Робота с Поля.
   3. **По указанию игры** Большой робот перемещается на соседнюю ячейку по горизонтали в направлении к маленькому роботу.

**Пока** Маленький робот находится на поле и Большой Робот не поймал Маленького Робота.

1. Игра считает победителем Маленького Робота, который достиг Точки Выхода.
2. **Сценарий завершается**.

**2) Дочерний сценарий «**Лабиринт создаёт и размещает на Поле Стены, Маленького Робота, управляемого человеком, и Большого Робота, управляемого компьютером, Точку Выхода, размещает Болота**»**

1. Лабиринт создаёт и расставляет последовательности Стен по периметру Поля.
2. Лабиринт создаёт и расставляет последовательности Стен внутри Поля.
3. Лабиринт создаёт Маленького Робота.
4. Лабиринт создаёт Большого Робота.
5. Лабиринт помещает на Поле созданных Роботов.
6. Лабиринт создает и размещает в ячейке Точку Выхода.
7. Лабиринт размещает в Ячейках Болота.
8. Сценарий завершается.

**3) Дочерний сценарий** «Маленький Робот перемещается на соседнюю Ячейку»

1. **В ответ на запрос** МаленькогоРобота Поле **сообщает**, что Стены нет в направлении движения Робота.
2. **В ответ на запрос** МаленькогоРобота Поле **сообщает**, что Ячейки с Болотом нет в направлении движения Робота.
3. Робот решает, что он может переместиться в соседнюю Ячейку.
4. Робот меняет свои координаты.
5. **Сценарий завершается**.

**4) Дочерний сценарий** «Большой Робот перемещается на соседнюю Ячейку»

1. Если счетчик пропущенных ходов равен не равен 0, Большой Робот пропускает ход и уменьшает счетчик на 1, иначе:
2. **В ответ на запрос** БольшогоРобота Поле **сообщает**, что Стены нет в направлении движения Робота, если Маленький Робот на Поле.
3. Большой Робот решает, что он может переместиться в соседнюю Ячейку по горизонтали в направлении Маленького Робота.
4. Большой Робот меняет свою позицию.
5. Если Большой Робот находится на Ячейке с Болотом и счетчик пропущенных ходов равен нулю, счетчик пропущенных ходов становится равен 3.
6. **Сценарий завершается**.

**3.1) Альтернативный сценарий** «Между соседней Ячейкой и текущей находится Стена. Маленький Робот». Сценарий **выполняется с** п. 1 сценария 3

1. **В ответ на запрос** Робота Поле **сообщает**, что Стена имеется в направлении движения Маленького Робота.
2. Робот решает, что он не может переместиться в соседнюю Ячейку.
3. **Сценарий завершается**.

**4.1) Альтернативный сценарий** «Между соседней Ячейкой и текущей находится Стена. Большой Робот». Сценарий **выполняется с** п. 1 сценария 4

1. **В ответ на запрос** Робота Поле **сообщает**, что Стена имеется в направлении (вправо или влево в сторону Маленького Робота) движения Большого Робота.
2. Робот решает, что он не может переместиться в соседнюю Ячейку.
3. **В ответ на запрос** Робота Поле **сообщает**, имеется ли Стена в направлении (верх или вниз в сторону Маленького Робота) движения Большого Робота.
4. Большой Робот меняет свою позицию, если стены нет.
5. **Сценарий завершается**.

**5) Дочерний сценарий** «Игра считает победителем Маленького Робота, который достиг Точки Выхода»

1. **В ответ на запрос** Игры Поле проверяет наличие Маленького Робота на себе.
2. Если Маленького Робота на поле нет, Игра считает победителем Пользователя.
3. **Сценарий завершается.**

**6) Дочерний сценарий** «Точка Выхода сообщает Игре, что Маленький Робот достиг ее, что приводит к исчезновению Робота с Поля»

1. Точка Выхода сообщает игре Маленький Робота достиг ее.
2. В ответ на запрос Точки Выхода, Игра убирает с Поля Маленького Робота.
3. **Сценарий завершается.**

**5.1) Альтернативный сценарий** «Проигрыш, Большой Робот поймал Маленького Робота».

1. Большой Робот **сообщает** Игре, что он «поймал» Маленького. (находится на той же позиции, что и Маленький Робот)
2. Игра считает, что Пользователь проиграл.
3. **Сценарий завершается.**

**6) Альтернативный сценарий** «Досрочное завершение игры». Сценарий **выполняется в любой точке** главного сценария

1. **По указанию пользователя,** программа завершается без определения победителя.
2. **Сценарий завершается.**

### 3.3 Словарь предметной области

*знает/умеет/предназначена*

**Игра** - знает о Поле и Лабиринте. Игра инициирует создание Поля и расстановку всех сущностей на нем с помощью Лабиринта. Игра определяет очередного активного Игрока и окончание игры (и победителя).

**Поле** - прямоугольная область, состоящая из Ячеек. Между Ячейками может располагаться Стена. По границе Поля также могут располагаться Стены. Знает о Роботах, находящихся на Поле, и Роботах, “телепортированных” с Поля.

**Позиция** – два числа, которые задают местоположение объекта на поле.

**Стена** - непроходимое Препятствие для Робота, располагающееся между Ячейками.

**Лабиринт** - умеет создавать Стены, Роботов и размещать их на Поле в Ячейках. Наполняет Ячейки Болотом. Позиции этих сущностей Лабиринт определяет самостоятельно.

**Робот** - умеет однократно *перемещаться* в течение своего хода. Робот перемещается в соседнюю Ячейку, но не может пройти через Стену.

**Маленький Робот** - Робот, который находится под управлением Пользователя.

**Большой Робот** - Робот, который находится под управлением Компьютера.

**Точка Выхода** - разновидность Ячейки. Она сообщает о Маленьком Робота, достигшим нее, и инициализирует его удаление с поля. На Поле может быть только одна Точка Выхода.

### 3.4 Структура программы на уровне классов

Диаграмма классов вычислительной модели

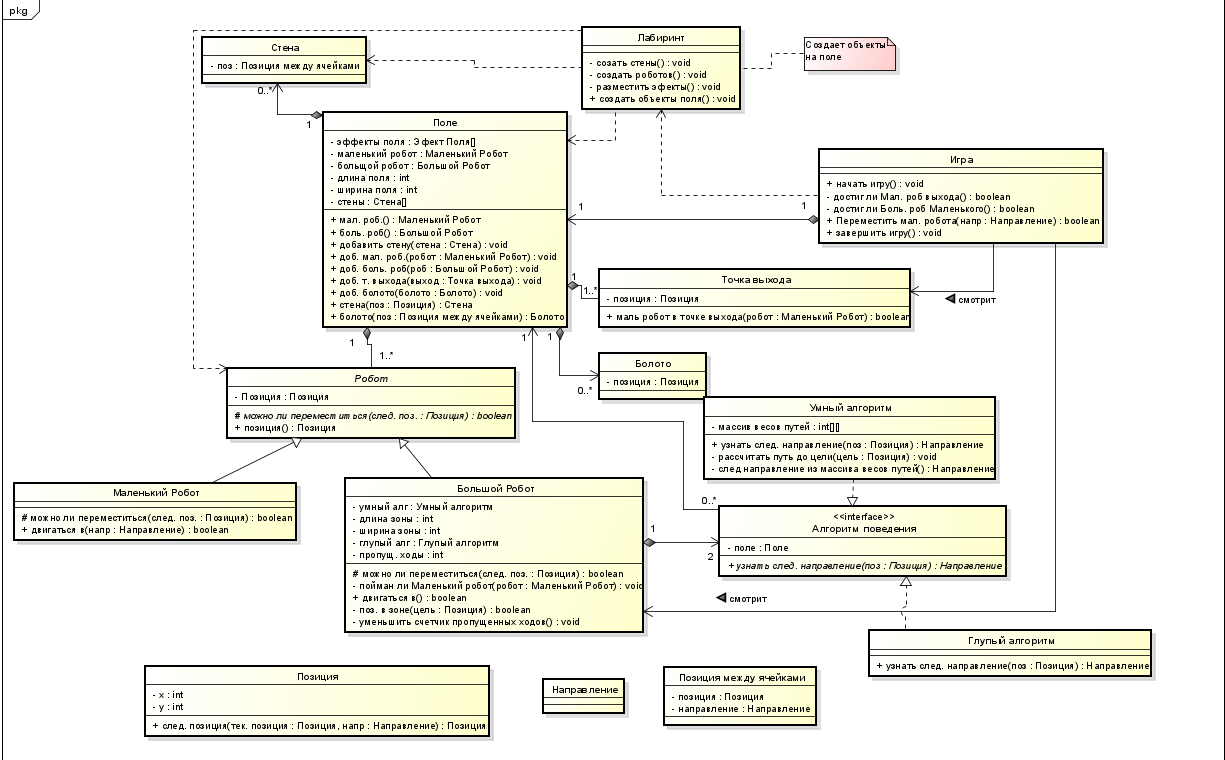
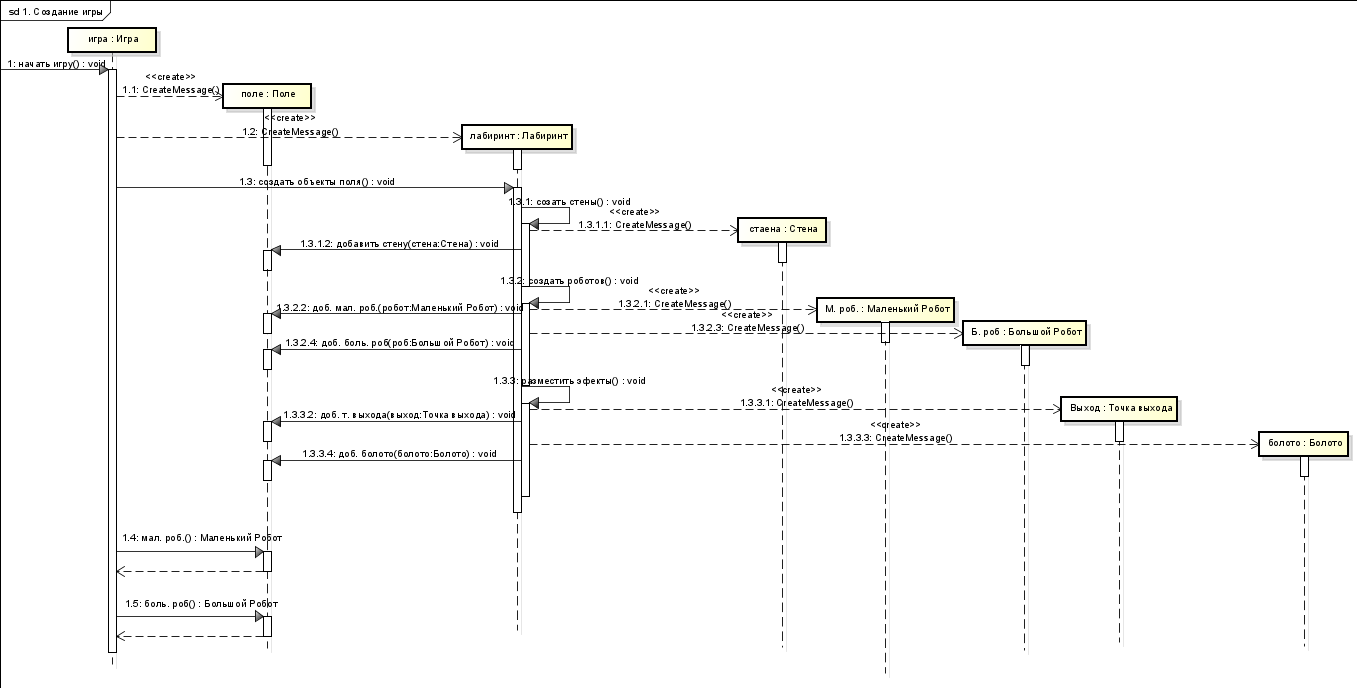


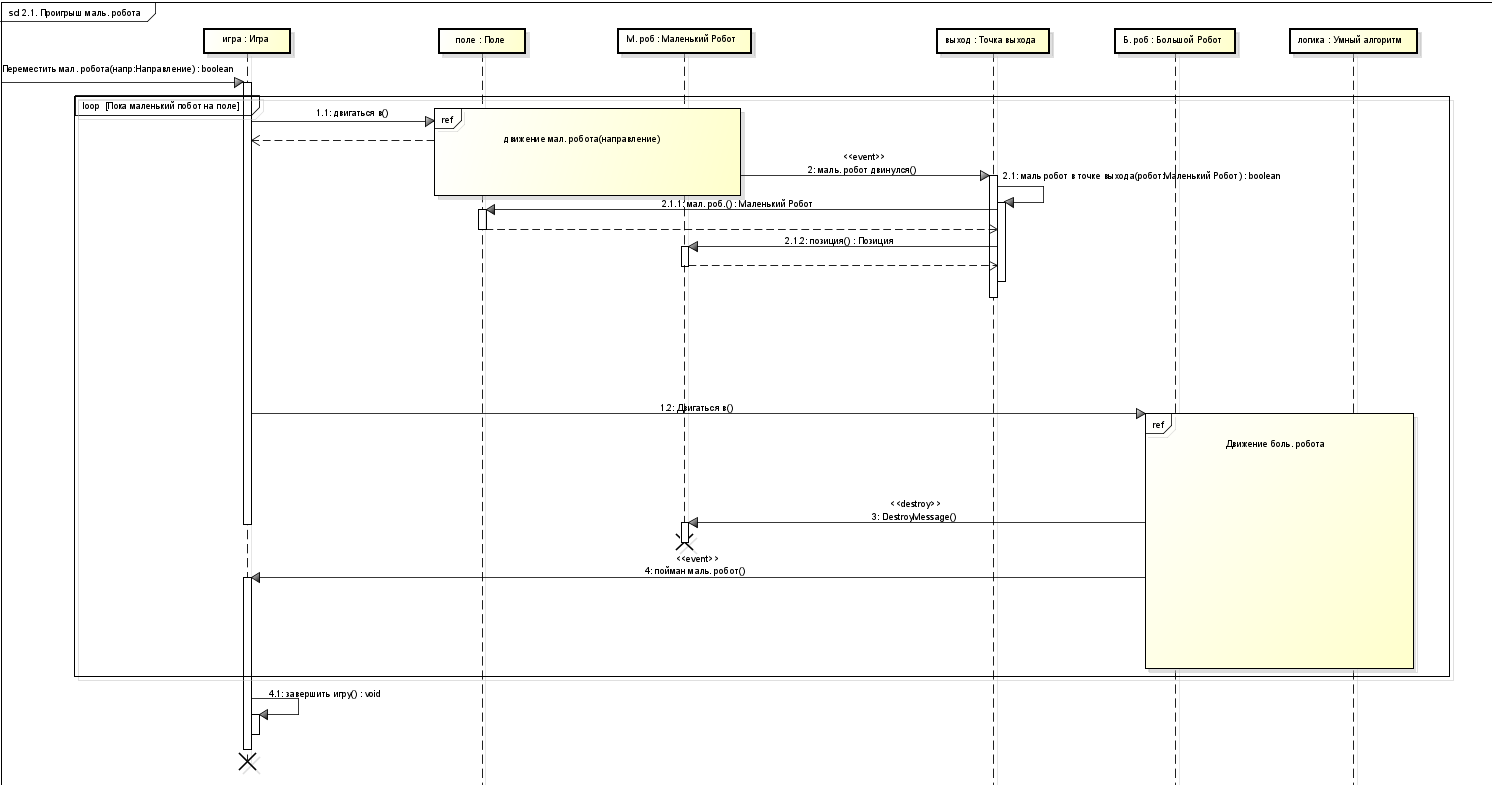
Диаграмма классов представления

### Типовые процессы в программе

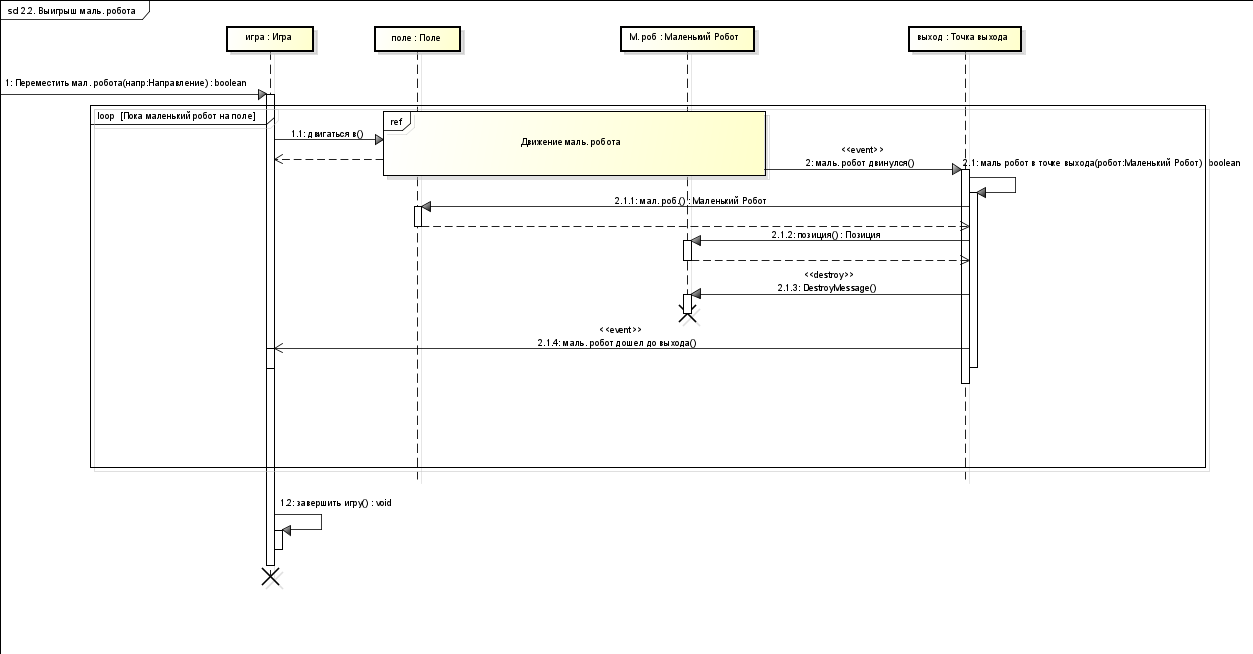
Начало игры:



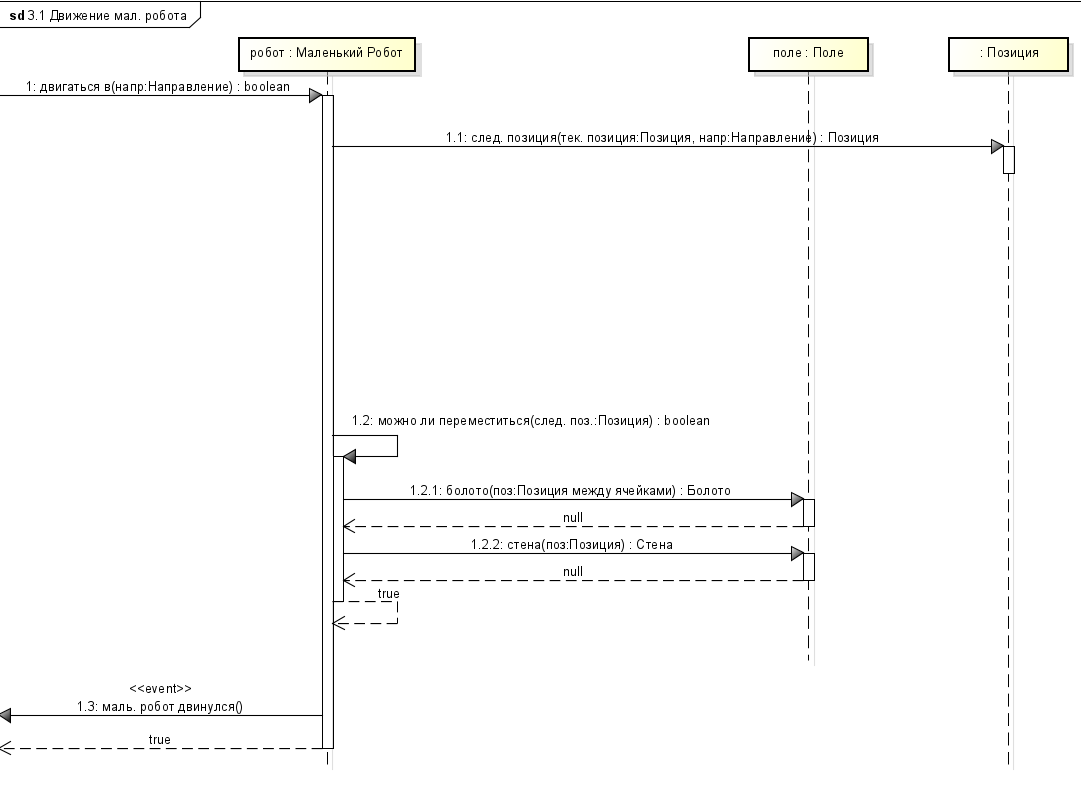
Проигрыш маленького робота



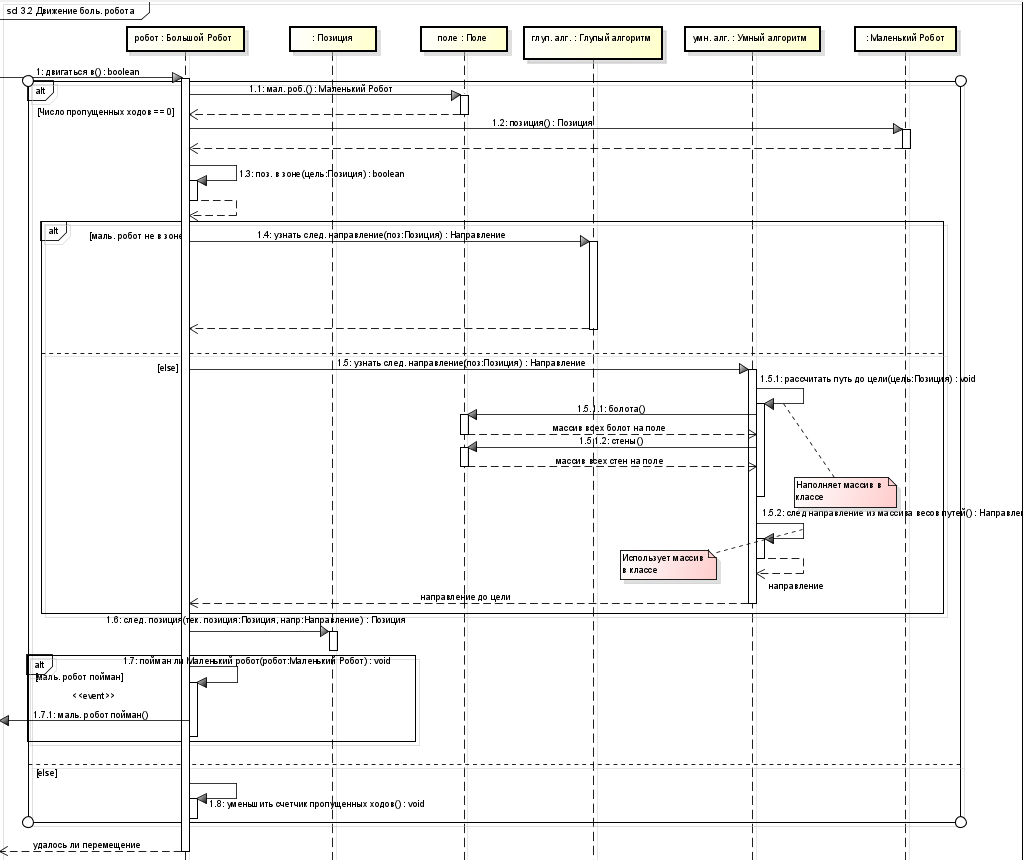
Выигрыш маленького робота



Движение маленького робота



Движение большого робота



### 3.6 Человеко-машинное взаимодействие

Общий вид главного экрана программы представлен ниже. На нём располагается игровое поле, на котором изображено игрок (маленький робот) и бот (большой робот) сверху и снизу, стены, болота и точка выхода.

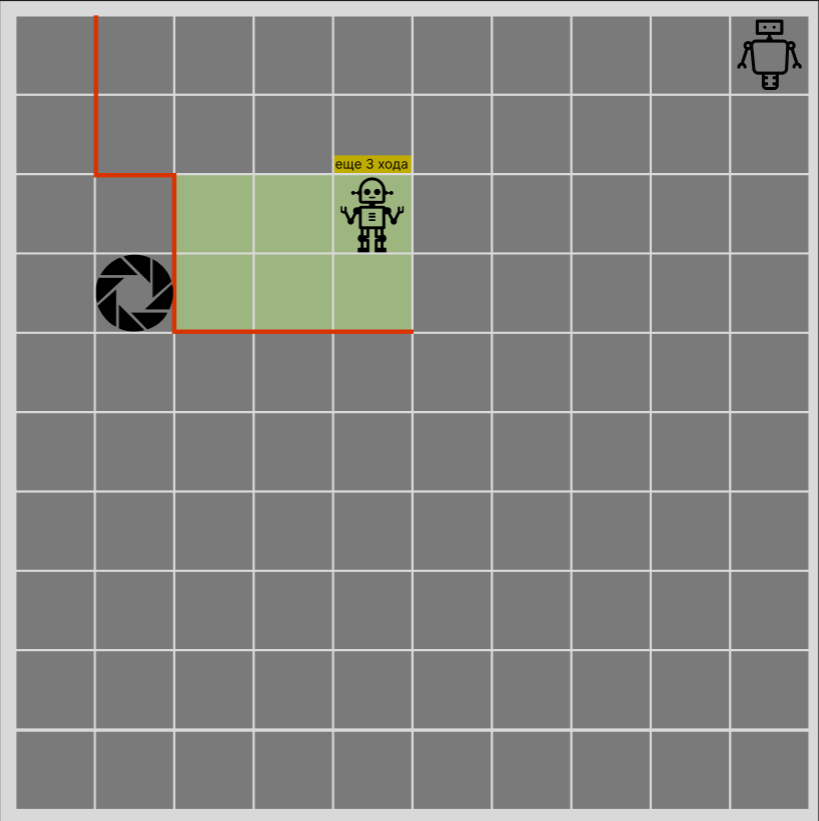


Рис. 1. Общий вид главного экрана программы

Управление активным роботом пользователь осуществляет с помощью клавиатуры.

W – движение вверх.

S – движение вниз.

A – движение влево.

D – движение вправо.

Изображение маленького робота представлено на рисунке 2, изображение большого робота на рисунке 3.

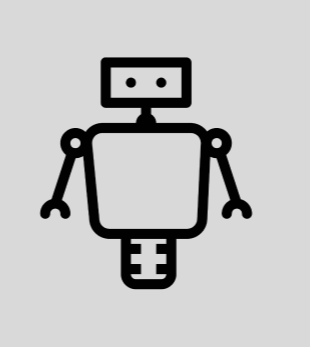


Рис.2. Маленький робот.

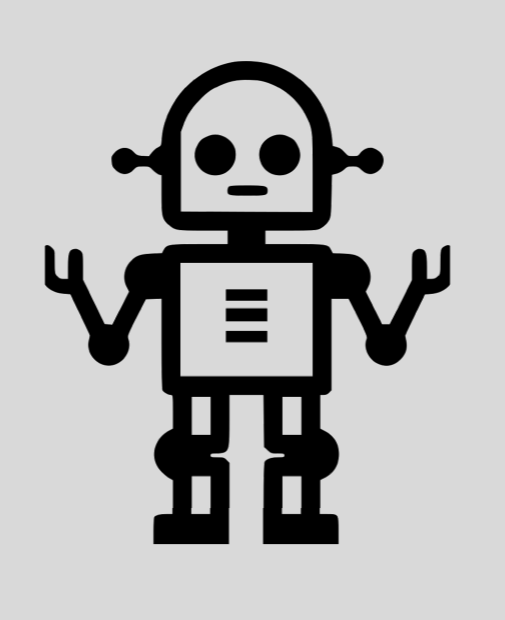


Рис. 3. Большой робот.

Кроме роботов на поле располагаются ячейка с выходом. Оно изображено на рисунке 4.



Рис. 4. Точка выхода.

Большой робот может попасть в болота, что заставит его пропустить 3 хода. Счетчик пропущенных ходов отобразится над большим роботом.

Пример изображён на рисунке 5.

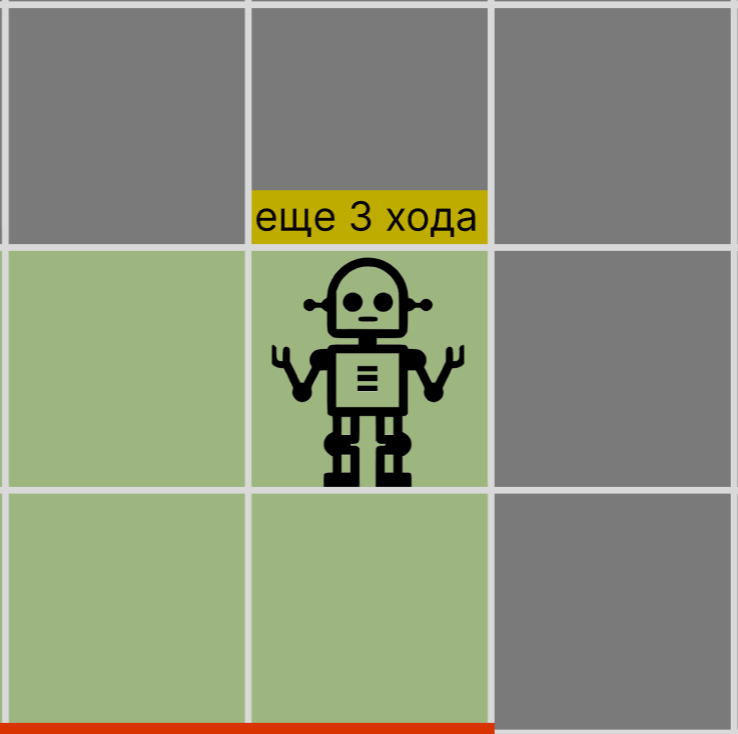


Рис. 5. Болото и счетчик пропущенных ходов.

Робот и батарейка могут располагаться в одной ячейке. В данном случае робот располагается слева, а уменьшенный вариант батарейки - справа. Пример расположения робота и батарейки в одной ячейке изображён на рисунке 6.



Рис. 6. Расположение робота и батарейки в одной ячейке.

При расположении робота и батарейки в одной ячейке у батарейки не отображается заряд, но его можно получить путём наведения на батарейку. Заряд батарейки отобразится в всплывающем окне. Пример отображение заряда представлен на рисунке 7.



Рис. 7. Заряд батарейки при расположении робота и батарейки в рамках одной ячейки.

### 3.7 Реализация ключевых классов

# 4 Вторя итерация разработки

### 4.1 Функциональные требования (сценарии)

### 4.2 Словарь предметной области

### 4.3 Структура программы на уровне классов

Диаграмма классов вычислительной модели

Диаграмма классов точки расширения

Диаграмма классов представления

### 4.4 Типовые процессы в программе

### 4.5 Человеко-машинное взаимодействие

Общий вид главного экрана программы представлен ниже. На нём располагается игровое поле, на котором изображено два игрока (робота) красный и синий, стены, двери, источники питания (батарейка, аккумулятор, ветряная мельница) и ячейка выхода (телепорт).

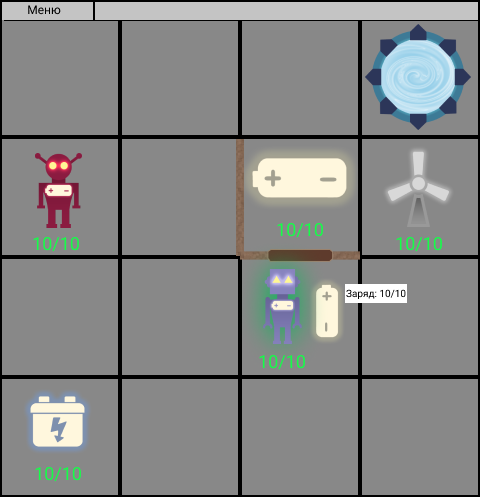


Рис.1. Главный экран программы.

Управление активным роботом пользователь осуществляет с помощью клавиатуры.

W – движение вверх.

S – движение вниз.

A – движение влево.

D – движение вправо.

G – сменить батарейку.

F – пропустить ход.

С – зарядить источник питания.

O – открыть/закрыть двери вокруг робота.

Изображение робота представлено на рисунке 2. Под изображением робота располагается надпись, информирующая о кол-ве заряда в роботе на данный момент. Робот также имеет два состояния. Неактивный робот изображён на рисунке 2. Активный робот изображён на рисунке 3. Активность робота показывается подсветкой зелёного цвета.

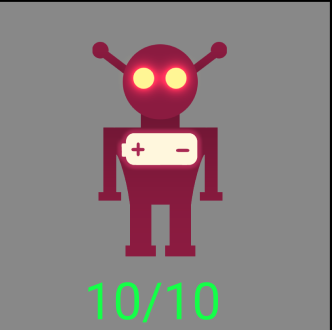


Рис.2. Робот.

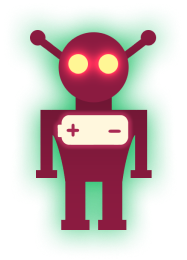


Рис. 3. Активный робот.

Кроме роботов на поле располагаются различные источники.

Батарейка – переносимый источник питания. Изображена на рисунке 4.

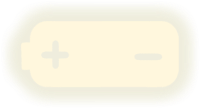


Рис. 4. Батарейка.

Аккумулятор – переносимый и перезаряжаемый источник питания. Изображён на рисунке 5.



Рис. 5. Аккумулятор.

Ветряная мельница – непереносимый возобновляемый источник питания.

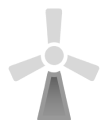


Рис. 6. Ветряная мельница.

Под изображением робота и батарейки располагается информация о кол-ве заряда, содержащегося в них. Цвет текста зависит от соотношения текущего заряда к максимально возможному.

* 0% - 30% - красный
* 30% - 70% - жёлтый
* 70% - 100% - зелёный

Пример изображён на рисунке 7.



Рис. 7. Цвета текста заряда робота и батарейки.

Робот и батарейка могут располагаться в одной ячейке. В данном случае робот располагается слева, а уменьшенный вариант батарейки - справа. Пример расположения робота и батарейки в одной ячейке изображён на рисунке 8.



Рис. 8. Расположение робота и батарейки в одной ячейке.

При расположении робота и батарейки в одной ячейке у батарейки не отображается заряд, но его можно получить путём наведения на батарейку. Заряд батарейки отобразится в всплывающем окне. Пример отображение заряда представлен на рисунке 9.



Рис. 9. Заряд батарейки при расположении робота и батарейки в рамках одной ячейки.

Помимо стен на поле имеются двери, которые могут иметь два состояния: Открытая (пример на рисунке 10) и закрытая (пример на рисунке 11)

.

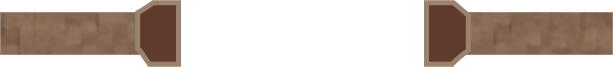


Рис. 10. Открытая дверь



Рис. 11. Закрытая дверь.

### 4.6 Реализация ключевых классов

# 5 Список использованной литературы и других источников

1. Логинова, Ф.С. Объектно-ориентированные методы программирования. [Электронный ресурс] : учеб. пособие — Электрон. дан. — СПб. : ИЭО СПбУТУиЭ, 2012. — 208 с. — Режим доступа: http://e.lanbook.com/book/64040
2. Васильев, А.Н. Самоучитель Java с примерами и программами. [Электронный ресурс] : самоучитель — Электрон. дан. — СПб. : Наука и Техника, 2016. — 368 с. — Режим доступа: http://e.lanbook.com/book/90231
3. Программирование на языке Java. Конспект лекций. [Электронный ресурс] : учеб. пособие / А.В. Гаврилов [и др.]. — Электрон. дан. — СПб.: НИУ ИТМО, 2015. — 126 с. — Режим доступа: http://e.lanbook.com/book/91488

**Перечень замечаний к работе**

1. Технические детали в интерфейсе - нарушение инкапсуляции - см. диаграмму классов
2. Человеко-машинное взаимодействие не всегда интуитивно-понятное