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 №17)

Студент: Ковалева А. 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: Ковалева А. А.

Группа: ПрИн-367

1. Тема: «Проектирование и реализация программы с использованием объектно-ориентированного подхода» (индивидуальное задание – вариант №17)

Утверждена приказом от « » 2023г. №

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

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 « » 2023г.

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

Задание принял к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ковалева А. А.

« » 2023г.

**Содержание**

1 Формулировка задания

2 Нефункциональные требования

3 Первая итерация разработки

3.1 Формулировка упрощённого варианта задания

3.2 Функциональные требования (сценарии)

3.3 Словарь предметной области

3.4 Структура программы на уровне классов

3.5 Типовые процессы в программе

3.6 Человеко-машинное взаимодействие

3.7 Реализация ключевых классов

3.8 Реализация ключевых тестовых случаев

**1 Формулировка задания**

**Игра "Танки".**

* Ира ведется пошагово двумя игроками;
* Игрок может пропускать свой ход, т.е. ничего не делать;
* На поле NxM клеток находятся танк игрока, танк противника, штабы и препятствия;
* Танк игрока имеет несколько жизней;
* Каждый танк может стрелять неограниченное количество раз, но не чаще 1 раза за N ходов;
* Цель игры - уничтожить танк противника ИЛИ штаб противника;
* Препятствиями могут быть:
  + кирпичная стена, которая может быть разрушена;
  + вода - это непроходимое препятствие.

**Дополнительные требования:**

* полет снаряда должен визуализироваться;
* разрушение объекта должно визуализироваться.

**Подвариант 2:** необходимо предусмотреть в программе **точки расширения**, используя которые можно реализовать вариативную часть программы (в дополнение к базовой функциональности).

**Вариативность**: предусмотреть возможность создания новых видов снарядов, отличающихся областью поражения, траекторией и дальностью полета.

**НЕ изменяя** ранее созданные классы, а используя **точки расширения**, **реализовать**: умный снаряд, который летит в заданную точку по кратчайшему маршруту (может облетать препятствия). Дальность полета снаряда ограничена. Область поражения - 4 смежных клетки.

**2 Нефункциональные требования**

1. Программа должна быть реализована на языке Java SE 12 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**

**Игра "Танки".**

* Игра ведется пошагово двумяигроками;
* Игрок может пропускать свой ход, т.е. ничего не делать;
* На поле NxM клеток находятся танк игрока, танк противника, штабы и препятствия;
* Танк игрока имеет несколько жизней;
* Каждый танк может стрелять неограниченное количество раз, но не чаще 1 раза за N ходов;
* Цель игры - уничтожить танк противника ИЛИ штаб противника;
* Препятствиями могут быть:
  + кирпичная стена, которая может быть разрушена;
  + вода - это непроходимое препятствие.

**Дополнительные требования:**

* полет снаряда должен визуализироваться;
* разрушение объекта должно визуализироваться.

**3.2 Функциональные требования (сценарии)**

1) Сценарий «Играть»

1. Пользователь инициирует начало игры

2. Игра создает при помощи Генерации поля Поле из Ячеек и размещает на нём два Танка и два Штаба к ним, Стены и Воду

3. Игра запрашивает у Поля Танки, которые находятся на нем

4. Игра случайным образом выбирает активный Танк

5. Делать

5.1. По указанию пользователя Танк перемещается на соседнюю Ячейку, стреляет в заданном направлении или пропускает ход

5.2. Игра запрашивает у Поля танки, которые находятся на нем

5.3. Игра делает активным следующий живой Танк, который располагается на Поле

Пока на поле есть хотя бы один живой Танк

6. Игра считает победителем единственный Танк, который уничтожил Танк противника или Штаб противника

2) Сценарий «Генерация поля создает поле из ячеек и размещает на нем два танка, два штаба, стены и лужи»

1. Игра инициирует создание поля размером NxM ячеек посредством генерации поля

2. Генерация поля создает и расставляет стены внутри поля

3. Генерация поля создает и расставляет воду внутри поля

4. Генерация поля создает два танка и помещает их на поле

6. Генерация поля создает и помещает два штаба на поле

3) Сценарий «Танк стреляет»

1. Пользователь хочет стрельнуть в заданном направлении

2. Танк разрешает себе стрельнуть (так как стрелять можно 1 раз за 3 хода)

3. Танк стреляет в заданном направлении

4. Танк обнуляет себе возможность стрелять

5. Снаряд летит до первой несвободной ячейки или края поля

6. Снаряд уничтожает препятствие, а потом и себя

7. Препятствием оказывается штаб, поэтому игра заканчивается с определением победителя

3.1) Альтернативный сценарий «Танк не может стрельнуть»

1. Сценарий начинается после пункта 1

2. Танк запрещает себе стрельнуть (так как стрелять можно 1 раз за 3 хода) и сообщает об этом пользователю

3. Сценарий переходит к пункту 5.1 главного сценария

3.2) Альтернативный сценарий «Снаряд попадает в стену периметра»

1. Сценарий начинается после пункта 2

2. Снаряд понимает, что перед ним ячеек больше нет, и саморазрушается

3. Сценарий переходит к пункту 5.2 главного сценария

3.3) Альтернативный сценарий «Снаряд попадает в стену»

1. Сценарий начинается после пункта 7

2. Препятствием оказывается стена, поэтому снаряд ее разрушает

3. Сценарий переходит к пункту 5.2 главного сценария

3.4) Альтернативный сценарий «Снаряд попадает в танк»

1. Сценарий начинается после пункта 7

2. Препятствием оказывается танк, поэтому снаряд отнимает у него одну жизнь

3. Сценарий переходит к пункту 5.2 главного сценария

4) Сценарий «Танк перемещается на свободную ячейку»

1. Танк запрашивает у ячейки, в которой он находится, соседнюю ячейку в направлении своего движения

2. Ячейка сообщает о ячейке, с которой соседствует

3. Танк спрашивает у соседней ячейки, есть ли в ней непроходимое препятствие

4. Танк просит ячейку, в которой он находится, изъять его из нее

5. Ячейка извлекает танк из себя

6. Танк просит соседнюю ячейку поместить себя в нее

7. Ячейка помещает танк в себя, т.к. в ней нет другого танка

4.1) Альтернативный сценарий «В соседней ячейке находится непроходимое препятствие»

1. Сценарий начинается после пункта 3

2. Соседняя ячейка сообщает, что препятствие присутствует

3. Сценарий переходит к пункту 5.1 главного сценария

4.2) Альтернативный сценарий «Пропуск хода»

1. Сценарий выполняется вместо сценария 4

2. Пользователь инициирует пропуск хода

3. Сценарий переходит к пункту 5.2 главного сценария

5) Сценарий «Досрочное завершение игры»

1. Сценарий начинается в любой точке главного сценария, когда пользователь инициирует завершение игры

2. Игра завершается без определения победителя

6) Сценарий «Снаряд летит по свободной ячейке»

1. Снаряд запрашивает у ячейки, в которой он находится, соседнюю ячейку в направлении своего движения

2. Ячейка сообщает о ячейке, с которой соседствует

3. Снаряд спрашивает у соседней ячейки, есть ли в ней непроходимое препятствие

4. Снаряд просит ячейку, в которой он находится, изъять его из нее

5. Ячейка извлекает снаряд из себя

6. Снаряд просит соседнюю ячейку поместить себя в нее

7. Ячейка помещает снаряд в себя, т.к. в ней нет другого непроходимого объекта

**3.3 Словарь предметной области**

**Игра** знает о поле. Игра управляет игровым циклом: определяет очередного игрока, определяет окончание, определяет победителя.

**Генерация поля** создает поле и определяет начальную расстановку игровых элементов (танки, стены, лужи, штабы) на поле.

**Поле** - прямоугольная область, состоящая из ячеек. Позволяет получить танки, находящиеся на поле.

**Ячейка** - квадратная область поля. Знает о четырёх соседних ячейках. В ней может находиться по отдельности стена, лужа, штаб, танк. Она понимает, какой объект в ней находится. Ячейка может передать объекту, что его поразили.

**Стена** – разрушаемый объект одним снарядом, заполняет всю ячейку. Танк не может ее пройти, если не разрушит.

**Вода** – непроходимый объект, заполняет всю ячейку.

**Штаб** – разрушаемый объект одним снарядом, заполняет всю ячейку. После разрушения штаба игра заканчивается.

**Снаряд** – объект, которым стреляет танк в выбранном направлении. Летит до первого препятствия. Сообщает ячейке с препятствием, что он ее поразил.

**Танк** – объект игры, который может перемещаться в соседнюю ячейку. Также умеет стрелять снарядом 1 раз в 3 хода. Не может переходить стены и лужи. Танк может пропустить ход и остаться в исходной ячейке.

**Активный танк** – танк, который может совершать действие в текущий ход.

**Направление** – класс для обозначения направления перемещения танка. Выбирается пользователем с клавиатуры.

**3.6 Человеко-машинное взаимодействие**

Общий вид главного экрана программы представлен ниже. На нём располагается игровое поле, на котором изображено два танка и два штаба (зеленый и синий), стены, вода. Каждый объект занимает одну ячейку.

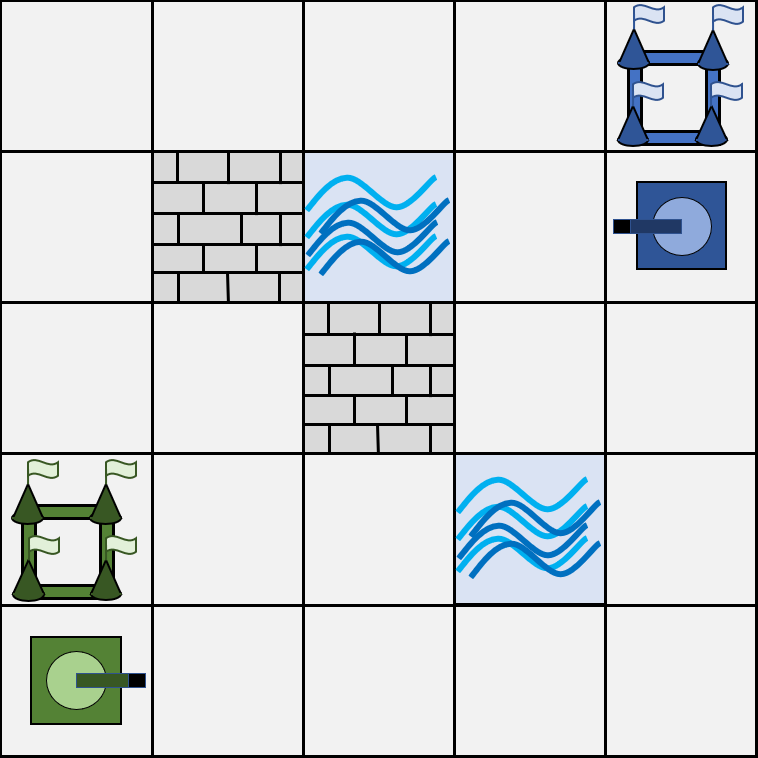


Рис. 1. Общий вид главного экрана программы

Управление активным танком пользователь осуществляет с

помощью клавиатуры.

W – направление вверх.

S – направление вниз.

A – направление влево.

D – направление вправо.

Space - идти.

F – стрелять.

Z – пропустить ход.

Изображение танка представлено на рисунке 2. Его дуло повернуто в ту сторону, которую он стреляет или ходит. Стрелять он может только один раз в 3 своих хода. Танк через все занятые клетки пройти не может.

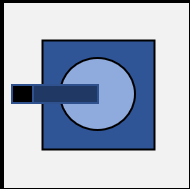


Рис.2. Танк

Также у танка есть штаб того же цвета, что и танк. Он представлен на рисунке 3.

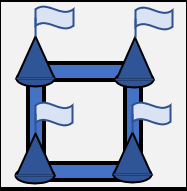


Рис.3. Штаб.

На поле есть препятствия. На рисунке 4 показана стена, которую можно разрушить одним выстрелом, а на рисунке 5 показана вода, ее разрушить нельзя.

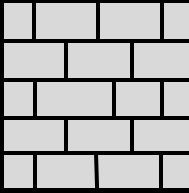


Рис.4. Стена.



Рис.5. Вода.

Танк стреляет снарядом, представленном на рисунке ниже. Он появляется в соседней ячейке, куда направлен танк.

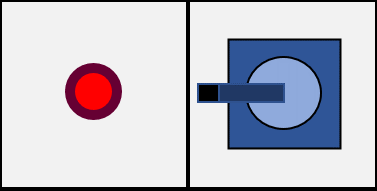


Рис.6. Танк стреляет снарядом.

Снаряд пролетает над водой, но разрушает стену и штаб, а также наносит урон в виде отнимания одной жизни у танка. Пример полета снаряда и его взрывы ниже.



а)



б)



в)

Рис.7. Полет снаряда, его взрыв и исчезание объекта.