МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Объектно-ориентированное программирование»

Тема: **Интерфейсы, динамический полиморфизм**

Студент гр. 1384 Алиев Д. А.

Преподаватель Жангиров Т. Р.

Санкт-Петербург
2022

Задание.

Реализовать систему событий. Событие - сущность, которая срабатывает при взаимодействии с игроком. Должен быть разработан класс интерфейс общий для всех событий, поддерживающий взаимодействие с игроком. Необходимо создать несколько групп разных событий реализуя унаследованные от интерфейса события (например, враг, который проверяет условие, будет ли воздействовать на игрока или нет; ловушка, которая безусловно воздействует на игрока; событие , которое меняет карту; и.т.д.). Для каждой группы реализовать конкретные события, которые по разному воздействуют на игрока (например, какое-то событие заставляет передвинуться игрока в определенную сторону, а другое меняет характеристики игрока). Также, необходимо предусмотреть событие “Победа/Выход”, которое срабатывает при соблюдении определенного набора условий.

Реализовать ситуацию проигрыша (например, потери всего здоровья игрока) и выигрыша игрока (добрался и активировал событие “Победа/Выход”)

Требования:

- Разработан интерфейс события с необходимым описанием методов
- Реализовано минимум 2 группы событий (2 абстрактных класса наследников события)
- Для каждой группы реализовано минимум 2 конкретных события (наследники от группы события)
- Реализовано минимум одно условное и безусловное событие (условное - проверяет выполнение условий, безусловное - не проверяет).
- Реализовано минимум одно событие, которое меняет карту (меняет события на клетках или открывает расположение выхода или делает какие-то клетки проходимыми (на них необходимо добавить события) или не непроходимыми
- Игрок в гарантированно имеет возможность дойти до выхода

Примечания:

- Классы событий не должны хранить никакой информации о типе события (никаких переменных и функций дающих информацию о типе события)
- Для создания события можно применять абстрактную фабрику/прототип/строитель

Выполнение работы.

Для выполнения лабораторной работы был создан интерфейс Event, являющийся классом родителем для все его наследников событий. Также был создан класс StateMediator для контроля над состоянием игры в зависимости от срабатывай соответствующих событий абстрактного класса EventState, а также множество других событий.

Описание классов:

1. Event – класс интерфейс для подклассов событий, имеет одну чистую виртуальный метод execute, которая переопределяется от класса к классу. Имеет чистый виртуальный метод execute(Controller& ctrl), передающий ссылку на контроллер, что позволяет событиям манипулировать состояниями и полями в игре. От него наследуются три группы событий.

2. PlayerEvents – абстрактный класс, наследники которого совершают действия с игроком.

- EventHeal – лечит игрока на 10 единиц при наступлении на клетку к которой прикреплён.
- EventHurt – наносит игроку урон в размере 50 единиц, когда тот встаёт на клетку, к которой прикреплено это событие.
- EventGetWeight – увеличивает характеристику веса игрока на 10 единиц.
- EventLoseWeight – уменьшает вес игрока на 10 единиц.

3. EventState – абстактный класс наследники которого отвечают за состояние игры.

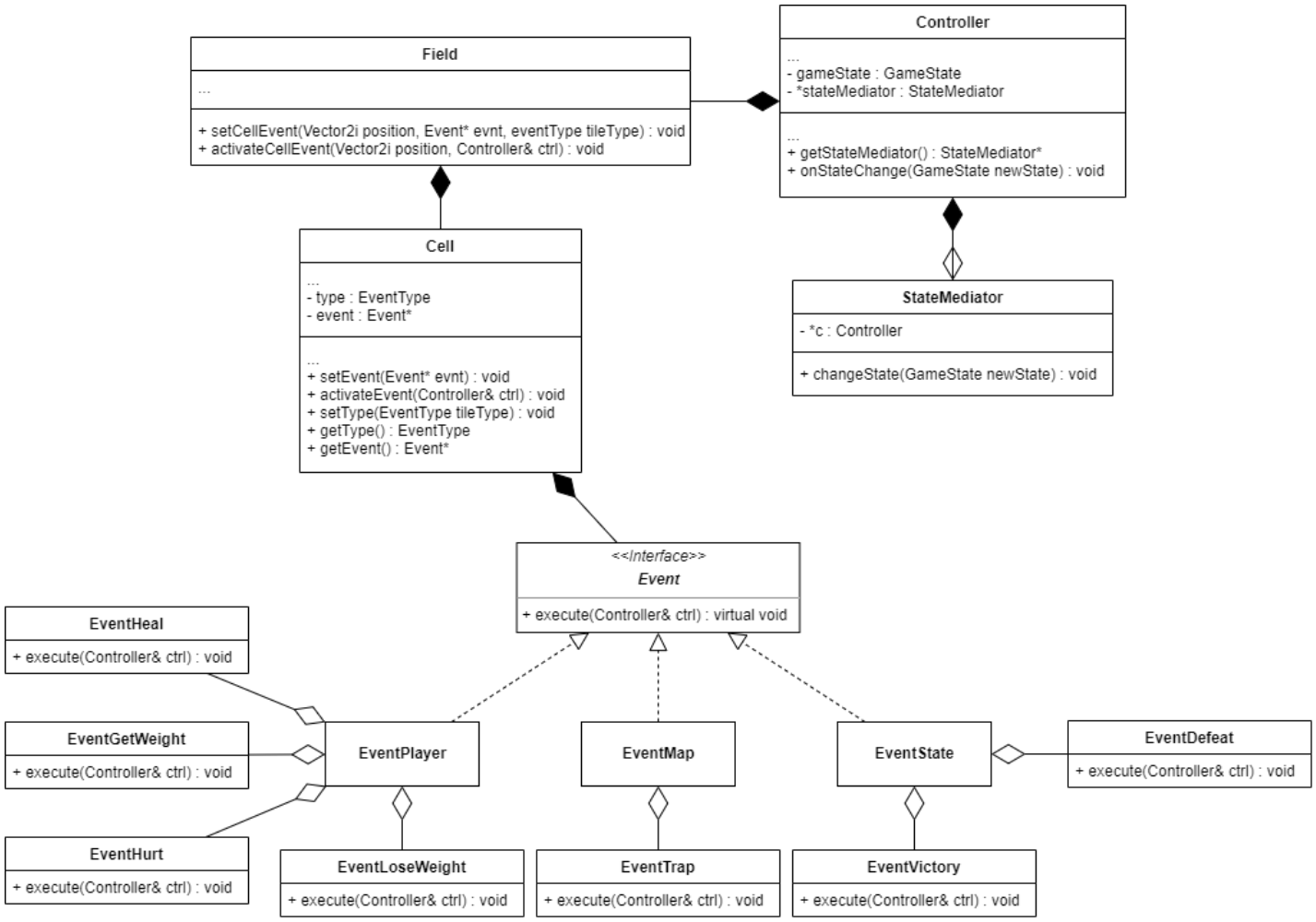
- EventVictory – условное событие, хранится в клетке, исполнимо только когда вес игрока равен 100.
- EventDefeat – условное событие, срабатывает при условии отсутствия здоровья у игрока или если тот наступит на клетку.

4. EventMap – абстрактый класс, наследники которого будут изменять карту.

- EventTrap – событие, которое заменяет клетки вокруг игрока на непроходимые, если тот наступит на клетку с событием.

5. StateMediator - класс, созданный для отслеживания и контроля передачи данных о событиях состояния игры. Меняет состояние игры на GameOver в случае срабатывания одного из событий состояния.

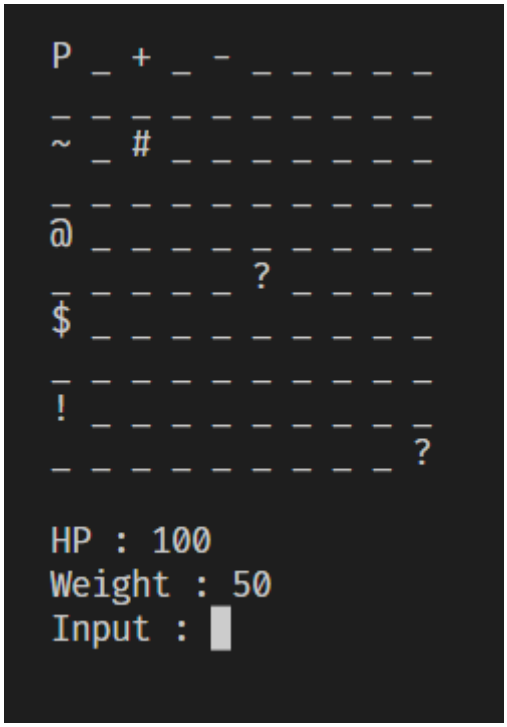
UML-диаграмма межклассовых отношений:



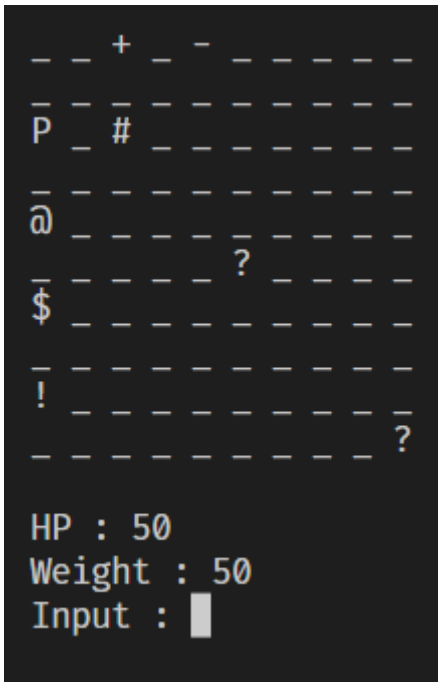
Тестирование программы :

Тестирование событий :

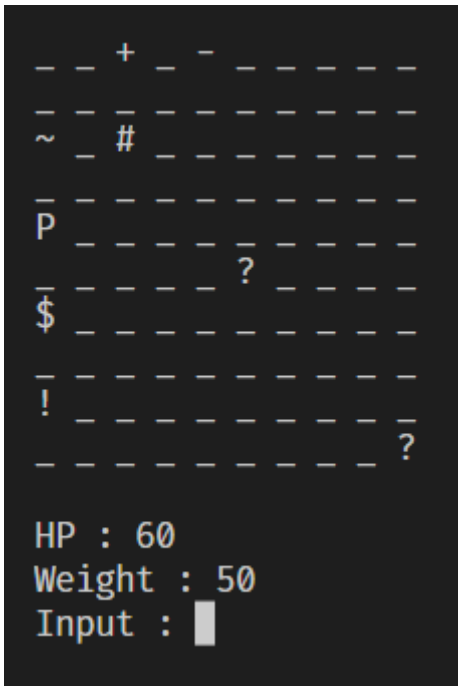
Карта изначально :



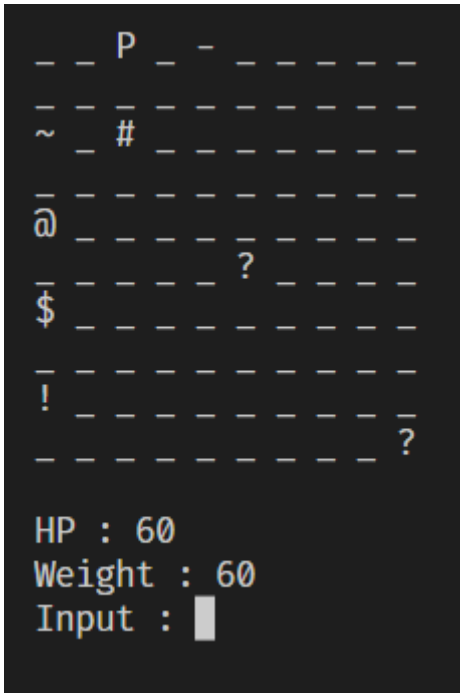
Урон ~:



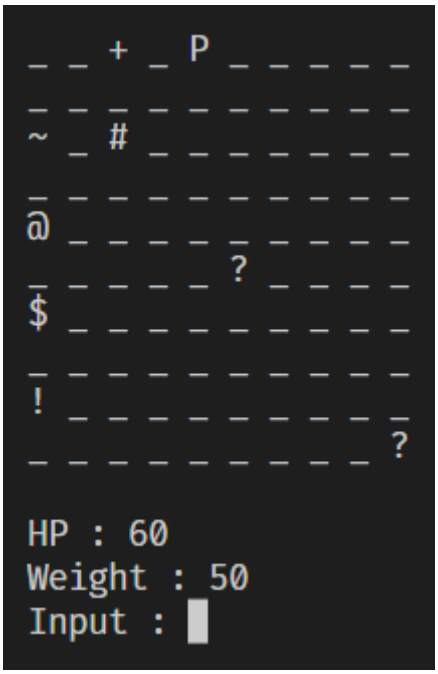
Лечение @:



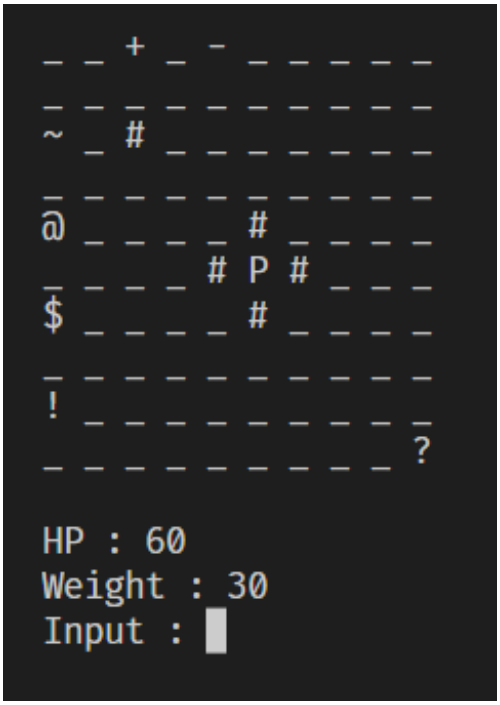
Вес + :

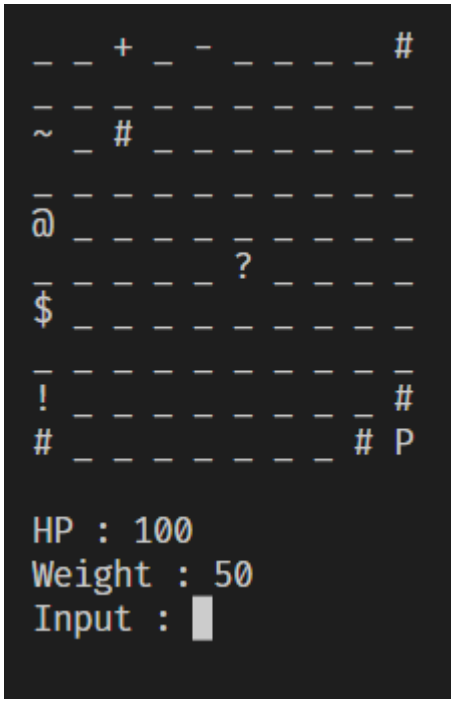


Вес – :

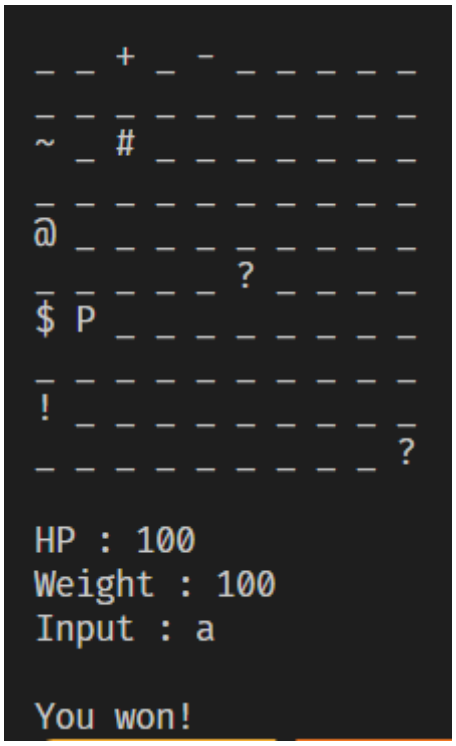


Ловушка ? (обычная и на границе карты):

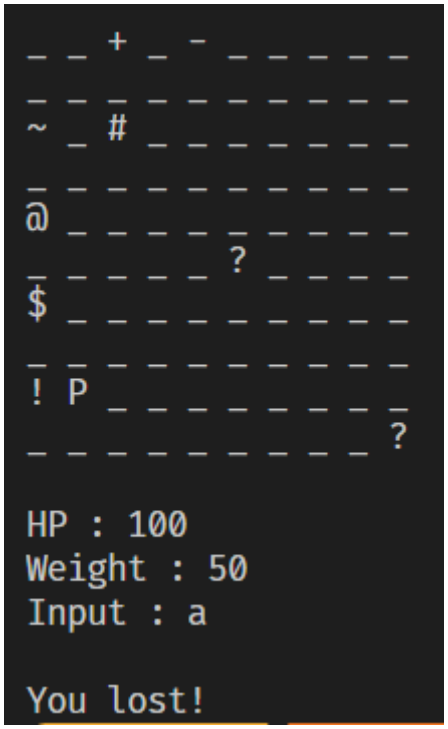


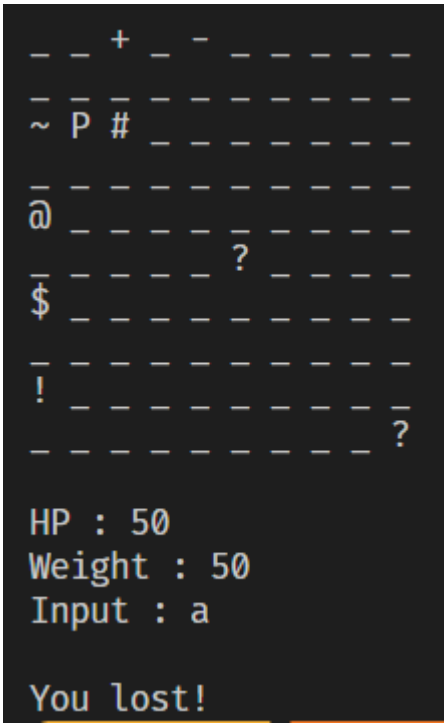


Победа $:



Поражение ! (при наступлении на клетку или потери всего здоровья) :





Вывод:

По результатам лабораторной работы были изучены интерфейсы и полиморфизм и реализована система событий.