

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

Прошичев А.В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2022

Цель работы.

Научиться работать с интерфейсами, сохранять полиморфизм и создавать множество наследуемых объектов, связанных в одну структуру.

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:

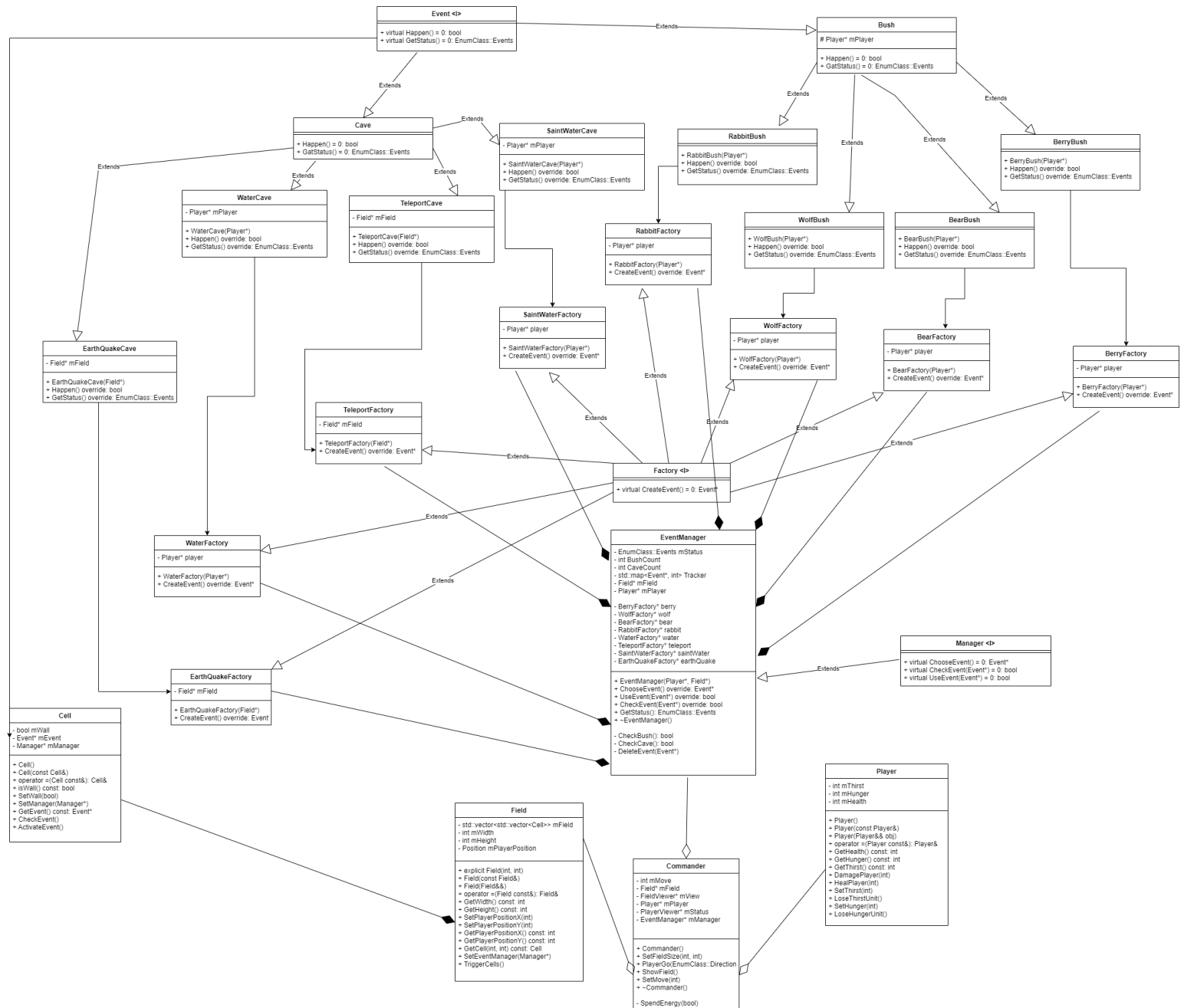
- Классы событий не должны хранить никакой информации о типе события (никаких переменных и функций дающих информацию о типе события)
- Для создания события можно применять абстрактную фабрику/прототип/строитель

Выполнение работы.

Главная задача лабораторной работы – создать структуру событий на карте. Интерфейсом любого событий является класс Event, он хранит в себе методы для взаимодействия с любым событием. Для управления событиями используется класс EventManager, реализация интерфейса Manager, – объект, позволяющий регулироваться работу всех событий Event. Хранится такой класс в классе Commander, и передаётся через класс Field всем классам клетки Cell. Теперь по поводу своего события клетка Cell можем обращаться к классу Manager.

Каждый ход класс Commander просит класс поля Field «пробежаться» по всем клеткам Cell и проверить наличие событий Event. Далее клетка обращается с соответствующим запросом выдать событие Event к классу EventManager. Класс EventManager хранит в себе фабрики, каждая из которых производит экземпляры определённого чёткого события: WaterCave (событие, утоляющее жажду игрока), SaintWaterCave (событие, восстанавливающие здоровье игроку), TeleportCave (событие, перемещающие игрока на другую пещеру), EarthQuakeCave (событие, порождающее вокруг себя другие события пещеры), BerryBush (событие, утоляющие голод игрока), BearBush (событие, убивающее игрока), WolfBush(событие, наносящее урон игроку), RabbitBush(событие,

значительно утоляющее голод игрока). Каждый из событий действует либо на игрока, либо на поле. Каждая фабрика реализует интерфейс Factory.



Тестирование.

```
Your hungry: H H H H h
Your thirst: 0 0 0 0 0 0 0 0
Your health: <3 <3 <3 <3 <3
Status: Nothing happened!
/ - - - - - \
      O      @
      @ @    @
      P
      O
O  @    O    O
O      O    @
      O      @
/ - - - - - \
Enter the activity: _
```

Изображение №1. Поле с отображением событий

```
Your hungry: H H H H h
Your thirst: 0 0 0 0 0 0 0 0
Your health:
Status: A bear came out from the bush!
/ - - - - - \
      O      @
      P @    @
      @
      O
O  @    O    O
O      O    @
      O      @
/ - - - - - \
---DEFEAT---
Unfortunately, you're dead!
Thank you for the test! See you later :)
```

Изображение №2. Смерть

```
Your hungry: H H H H h
Your thirst: 0 0 0 0 0 0 0 0
Your health: <3 <3 <3 <3 <3
Status: You found such a delicious rabbit!
```

Изображение №3. Смена статуса игры.

```
Enter the activity: w
---VICTORY---
You survived. You won.
Thank you for the test! See you later :)
```

Изображение №4. Победа

Выводы.

Была освоена работа с интерфейсами. Создана структура из событий.
При реализации были освоены паттерны Фабрика и Фасад.