**Домашнее задание на практику №3.**

## Задание 1.

Настало время экспериментов! Создание игр подчиняется старой схеме, позаимствованной у великих иллюзионистов - дым и зеркала. И для придания иллюзии “живой” игры, необходимо сделать так, чтобы некоторые действия происходили параллельно, пока игрок продвигается по миру. А значит стоит задача внедрить многопоточность.

Для начала имеет смысл заселить игру существами, которые могут “передвигаться” независимо от действий пользователя.

Во время сражения, сила атаки существа рассчитывается через [DPS](#6qyqy7ak48wa).

*Пояснение: Данные для функции DPS должны браться из ячеек экипировки существа.*

Для управления существом, как отдельным потоком выполнения, используется класс CreatureController, реализующий интерфейс Runnable. Класс содержит в себе поле объекта Mob с ссылкой на конкретное существо и позволяет ему совершать действия в отдельном потоке, пока тот жив.

*Пояснение: Конструктор CreatureController принимает в себя объект Creature (а если быть точнее, ссылку на объект) и присваивает его внутреннему полю класса, чтобы далее использовать в своих методах.*

Что может делать существо в своём потоке: ожидать появление игрока и драться с игроком.

В качестве игрового пространства будет использоваться динамический массив объектов класса Cell - клетка.

Класс Cell должен содержать:

* Поле объекта класса Position, которое описывает пространственные координаты клетки на поле;
* Динамический массив объектов GameObject - объекты на клетке;
* Метод для получения текущего объекта на клетке;
* Метод размещения объекта на клетке;
* Метод для удаления объекта с клетки (таким образом, чтобы он не дублировался);

Поля класса должны быть приватными и не статическими.

Класс Position, содержит приватные поля x, y типа int а также дополнительными методами работы с ним.

Что необходимо сделать в задании, с учётом условий выше:

1. Изменить соответствующим образом классы существ, чтобы те содержали ссылку на клетку в которой они находятся (возможно лучше перенести поле клетки в класс Creature?)
2. В класс GameLogic внести статические методами для обработки различных игровых систем: расчёт DPS, бой, генерация оружия и т.д;
3. Класс Map, который будет содержать динамический массив клеток - собственно игровое поле. Класс должен содержать метод для генерации набора клеток, а так же поиск клетки по её координатам;
4. Каждый противник использует свой контроллер для работы в отдельном потоке и обладает своей экипировкой (оружие и броня);
5. Персонаж игрока *автоматически[[1]](#footnote-1)* путешествует по игровому полю и автоматически сражается с противником.
6. Перед каждым действием существа в контроллере происходит ожидание при помощи метода TimeUnit.MILLISECONDS.sleep(значение); Оно приостанавливает текущий поток на указанное количество миллисекунд. При этом в поле класса контроллера присутствует константное значение времени ожидания (в миллисекундах) существа. По-умолчанию оно может быть равно 1000, тем не менее, при желании его можно изменять для ускорение действий.

Способ генерации игровых переменных - рандомизация.

Игровой цикл состоит из следующих условий:

1. Все существа (игрок и противники) перемещаются в случайном направлении на одну клетку, при этом может возникнуть одна из ситуаций:
   1. Если на одной клетке встречаются два и больше противника - их базовая сила увеличивается на единичку;
   2. Если на одной клетке встречается игрок и один противник, то они сражаются;
   3. Если на одной клетке встречается игрок и несколько противников, то на битву выползает самый сильный из всех врагов (считается по DPS);
2. Если оружие или броня убитого противника лучше чем то, что носит игрок, то оно заменяет текущее;
3. Цикл заканчивается, если игрок погибает или не осталось противников на карте.

Примерный формат вывода логов в консоль:

Для движения по клеткам:

[MOB][ИМЯ\_МОБА]: перемещается с (x;y)(старые координаты) -> (x;y)(новые координаты)

[PLAYER][ИМЯ]: перемещается с (x;y) -> (x;y)(новые координаты)

Для битвы:

[PLAYER][ИМЯ]: наносит урон X ИМЯ\_МОБА

[MOB][ИМЯ\_МОБА]: наносит урон X герою ИМЯ

Если несколько мобов встретились на одной клетке:

[КЛЕТКА][X;Y]Встретились мобы: ИМЯ\_МОБА1, ИМЯ\_МОБА2 их базовая сила повысилась на 1 и стала N.

## 

## Задание 2.\*

(необязательное)

Реализовать собственный класс очереди (или стека) игровых объектов на клетке вместо динамического массива. Тип очереди LIFO. В качестве способа размещения объектов памяти можно использовать массив GameObject[].

Класс должен:

* Поддерживать итерацию объектов;
* Возвращать последний объект без удаления из очереди;
* Возвращать последний объект с удалением из очереди;
* Вставлять объект в очередь;
* При создании нового экземпляра очереди создавать массив с минимальным размером 10 элементов;
* Поддерживать динамическое расширение.

Решение задачи в зависимости от качества выполнения добавляет от 1 до 3 баллов (плюсиков) к общему решению практики.

## 

## 

## Что требуется в отчёте

1. Текст заданий;
2. Вывод данных по игровому циклу из консоли;
3. Листинг кода по каждому классу в конце отчёта;
4. Для задания 1 - диаграмма классов из IDE с полями и методами, а также линиями зависимостей.

## 

1. То есть, тоже в отдельном потоке. [↑](#footnote-ref-1)