**Финальное задание по курсу:**

**ООП и Паттерны проектирования**

**Для игры необходимо запустить main.py**

**Смерть героя: HERO Strength < Enemy strength**

Сила копится при поражении врага

Враги (enemies):

rat(strength: 2), knight (strength: 30), naga (strength: 80), dragon (strength: 300),

hydra (strength: 1000)

Управление:

" →", "Move Right"

" ←", "Move Left"

" ↑ ", "Move Top"

" ↓ ", "Move Bottom"

" H ", "Show Help"

"Num+", "Zoom +"

"Num-", "Zoom -"

" R ", "Restart Game"

Смысл задания был в том, чтобы разобраться в коде, как всё «вертится», что на что влияет, реализовать некоторые классы и методы и дописать недостающий код в некоторые из предоставленных функций (методов).

**Само задание приведено в конце файла (в том числе диаграмма классов).**

**Модуль main.py**

Основной запускающий файл игры. Здесь происходит запуск основных параметров игры, установка управления.

Также происходит запуск паттерна Цепочки обязанностей(drawer).

**Модуль logic.py**

Содержит логику игрового движка GameEngine, в котором описывается часть паттерна Наблюдатель (subscribe, unsubscribe, notify), взаимодействие игрока с другими объектами игры(interact), смещение игрока на карте при его перемещении(move\_up).

**Модуль ScreenEngine.py**

Данный модуль отвечает за отрисовку на экране всех основных элементов игры. Работа модуля осуществляется на основе поведенческого паттерна Цепочка обязанностей. Каждый обработчик отрисовывает свою часть и пeредаёт управление отрисовкой(draw) и движок(connect\_engine) нулевому обработчику, который передаёт следующему в цепочке.

Также в данном модуле реализован подписчик(InfoWindow) из паттерна Наблюдатель, который получает уведомление через метод update и выводит его на экран.

**Модуль Objects.py**

Данный модуль отвечает за создание героя(Hero), союзников(Ally), противников(Enemy), а также накладывает различные эффекты на героя(Effects).

**Модуль Service.py**

Данный модуль выполняет обслуживающие функции. Осуществляет создание начальных параметров игры, восстановления здоровья, применение blessing и berserk, снятие эффектов с героя, действие при взятии сундука, а также создание карт игры. Карты создаются с помощью порождающего паттерна Абстрактная фабрика (адаптированный к python с использованием @classmethod, что позволило сократить код)

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

НИЖЕ ПРИВЕДЕНО ЗАДАНИЕ КУРАТОРОВ КУРСА.

Задание: Игра «Рыцарь в подземелье»

Отправить до 10 февр. г., 11:59 SAMT

В данном задании Вам необходимо будет закончить разработку полноценной ролевой игры «Рыцарь в подземелье». В данной игре необходимо будет играть за рыцаря, который путешествует по многоэтажному подземелью, борется с врагами и собирает сокровища.

Вам будет дан код движка игры, текстуры и игровая логика. Некоторые из классов и методов изначально не реализованы в предоставленном коде. Их и необходимо будет реализовать в данном задании. В помощь вам будут даны полные диаграммы классов, которые должны быть представлены в проекте.

Кроме реализации новых методов, необходимо будет дописать недостающий код в некоторые из существующих функций и методов. Места, в которых требуется исправление, помечены меткой **#FIXME**.

Кроме основных заданий, Вам будут предложены дополнительные задания. В них вам нужно будет немного усовершенствовать игру. При выполнении дополнительных заданий старайтесь быть креативными.

В качестве ответа вам необходимо будет прикрепить архив, содержащий всю структуру папок Вашего проекта.

Успешного выполнения задания!

**Review criteria**

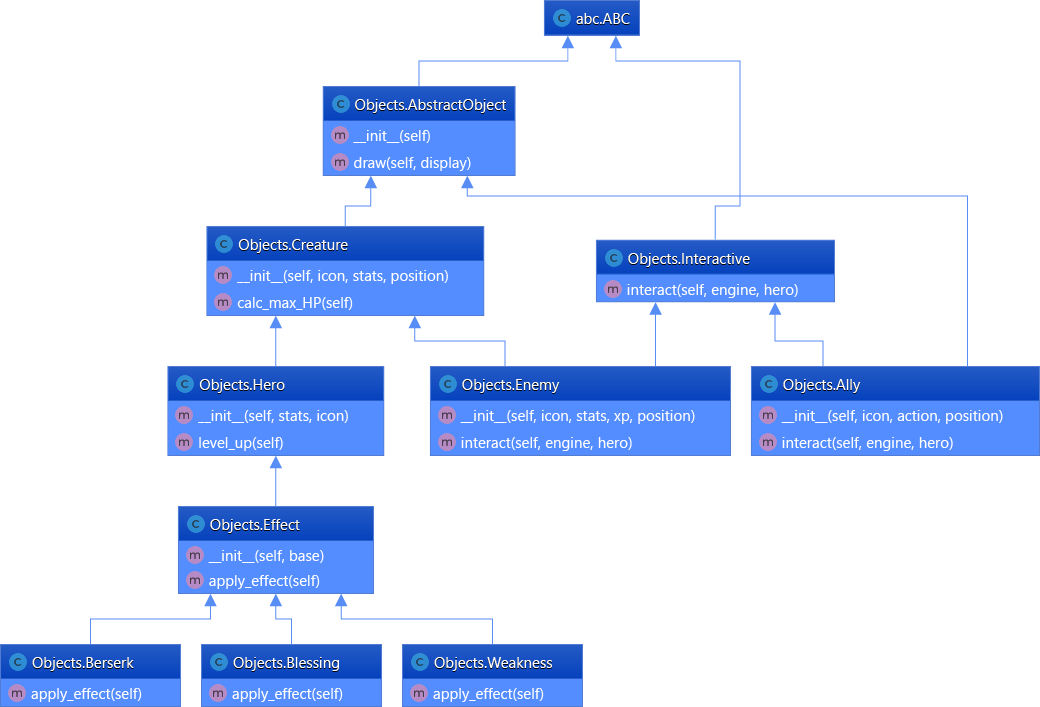
1. Должны быть реализованы все необходимые классы;
2. Схема наследования должна соответствовать заданной;
3. Программа должна корректно выполняться с исходными YAML файлами;
4. Реализация дополнительных заданий

Скачайте заготовку проекта:

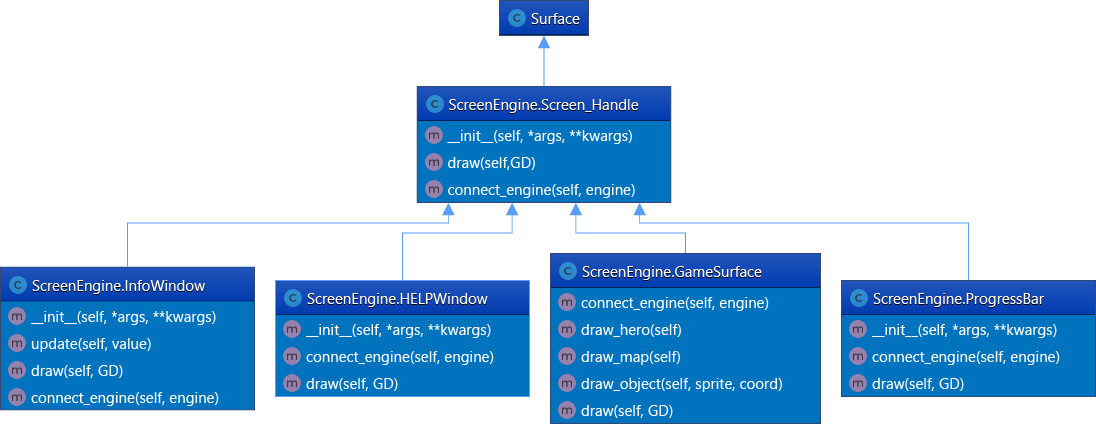
[final\_project.zip](https://d3c33hcgiwev3.cloudfront.net/YNVb7Vb9EeiOFxKmNjWWlA_613862c056fd11e8aabab58072b1f7dd_final_project.zip?Expires=1580169600&Signature=FdWlTRHjh9At49e2n7GlP4Vc4FJSr4esLpLb4wvb0SWjZySR1dg7RXx0MeMVAxcqLOXj~tvGbchhp-aBcOMZaJx6X~h10t0vl~3UTu06RLFLDCr725ORwXkVzkmyoVIOmJjdztJ~2wm6PgSDvt2K7RN~xvF5P5z48slCXYvNOlk_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

Реализуйте все классы в соответствии с UML диаграммами, и исправьте все #FIXME.

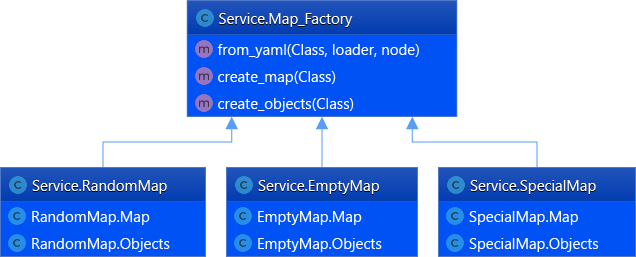
**Objects.py**



**ScreenEngine.py:**



**Service.py**:



**Дополнительные задачи:**

1. Реализованы собственный противник и союзник;
2. Добавлена отрисовка миникарты;
3. Реализован дополнительный эффект.