**Тестовое задание**

Данное задание максимально полно отражает типовые задачи на данной позиции. Цель задания - проверить владение синтаксисом и базовыми возможностями typescript, общее понимание принципов ООП и алгоритмов, а также навыки чистого кода.

Команда Banzai.Games желает вам удачи в его выполнении!

По сценарию задания вы получили от команды геймдизайна геймплейное описание, а ваш наставник дал вам набросок архитектуры и список простых задач. Изучите геймплейное описание, скопируйте листинг с архитектурой и дополните архитектуру до готового скрипта согласно списку задач. Задачи 9 и 10 выполнять желательно, но необязательно. Выполненное задание нужно отправить в одном из двух форматов на выбор:

- ссылка на публикацию на https://replit.com (желательно)\*;

- текстовый файл index.ts.

*\*Пример программы, опубликованной на replit:*

*https://replit.com/@GoriachievDmitrii/FatherlyMysteriousBusinesses#index.ts.*

*Для публикации может потребоваться регистрация.*

**Геймплейное описание:**

В нашей РПГ-игре персонаж игрока собирает в своём инвентаре предметы, полученные из разных источников наград.

Предмет может находиться в инвентаре в нескольких экземплярах.

Каждый предмет имеет следующие свойства:

● уникальный идентификатор **ID** (целое положительное число);

● название **name** (строка);

● редкость **rarity** (представлена перечислением **RARITY**);

● тип предмета **itemType** (представлена перечислением **ITEMTYPE**).

В игре 32 предмета, по 2 каждого типа каждой редкости.

Необходимо реализовать предметы, инвентарь и источник наград "бустерпак".

При открытии бустерпака игрок получает несколько предметов.

Бустерпаки могут иметь различную логику выдачи предметов:

1) Простой бустерпак **Booster**:

● выдаёт **N1** случайных предметов редкости **R** и **N2** предметов редкости **R-1** (например, 2 RARE и 3 COMMON).

2) Бустерпак удачи **LuckBooster** - то же, что **Booster**, но:

● каждый предмет имеет шанс **CK** замениться на предмет того же типа с редкостью на **K** единиц выше (например, должен был выпасть предмет редкости COMMON, шанс повышения редкости 10%. Значит с вероятностью 10% выпадет предмет редкости RARE, 1% - EPIC, 0,1% - LEGENDARY).

3) Равномерный бустерпак **UniformBooster** - то же, что **LuckBooster,** но:

● суммарно не менее 4 предметов;

● в выдаче должны присутствовать предметы всех типов.

4) Коллекционный бустерпак **CollectionBooster**- то же, что **UniformBooster ,** но:

● вероятность выпадения конкретного предмета должна быть обратно пропорциональна количеству экземпляров предмета в инвентаре (то есть чем больше в инвентаре экземпляров предмета M, тем меньше вероятность выпадения предмета M из этого бустерпака).

**Задачи:**

1) Дописать класс предмета **Item** и интерфейс его настроек.

2) Наполнить базу предметов **itemsBase** экземплярами класса **Item**.

3) Дописать класс бустерпака **Booster** и интерфейс его настроек.

4) Добавить **2** экземпляра класса **Booster** с разными настройками в коллекцию бустерпаков **boostersBase**:

● для выдачи **3** предметов **RARE** и **2** предметов **COMMON**;

● для выдачи **1** предмета **LEGENDARY** и **3** предметов **EPIC**;

5) Дописать класс бустерпака **LuckBooster**.

6) Добавить **2** экземпляра класса **LuckBooster** с разными настройками в коллекцию бустерпаков **boostersBase**:

● для выдачи **3** предметов **RARE** и **2** предметов **COMMON**, шанс повышения редкости **10%**;

● для выдачи **1** предмета **LEGENDARY** и **3** предметов **EPIC**, шанс повышения редкости **45%**;

7) Создать класс бустерпака **UniformBooster**.

8) Добавить **2** экземпляра класса **UniformBooster** с разными настройками в коллекцию бустерпаков **boostersBase**:

● для выдачи **3** предметов **RARE** и 2 предметов **COMMON**, шанс повышения редкости **10%**;

● для выдачи **1** предмета **LEGENDARY** и 3 предметов **EPIC**, шанс повышения редкости **45%**;

9)\* Создать класс бустерпака **CollectionBooster**.

10)\* Добавить 2 экземпляра класса **CollectionBooster** с разными настройками в коллекцию бустерпаков **boostersBase**:

● для выдачи **3** предметов **RARE** и **2** предметов **COMMON**, шанс повышения редкости **10%**; ● для выдачи **1** предмета **LEGENDARY** и **3** предметов **EPIC**, шанс повышения редкости **45%**;

11) Ниже комментария **// TEST** вызвать функцию **getBoosterLoot()** для каждого бустерпака в **boostersBase** по 1 разу, результаты вызовов вывести в консоль.

*\*Задачи 9 и 10 выполнять желательно, но не обязательно.*

**Архитектура:**

// Редкость

enum RARITY {

COMMON,

RARE,

EPIC,

LEGENDARY

}

// Тип предмета

enum ITEMTYPE {

HELMET,

WEAPON,

SHIELD,

ARMOR

}

// Настройки предмета

interface IItemSettings {

name: string;

}

// Класс предмета

class Item {

name: string;

constructor(settings: IItemSettings) {

this.name = settings.name;

}

}

// База предметов, ключ - ID предмета.

let itemsBase: {[ID: number]: Item} = {

1: new Item({name: "Shiny Sword"}), // пример добавления экземпляра предмета };

// Интерфейс инвентаря. Ключ - ID предмета, значение - количество экземпляров этого предмета в инвентаре

interface IInventory {

[key: number]: number

}

// Настройки простого бустерпака

interface IBoosterSettings {

rarity: RARITY;

}

// Класс простого бустерпака

class Booster {

rarity: RARITY;

constructor(settings: IBoosterSettings) {

this.rarity = settings.rarity;

}

getBoosterLoot(playerInventory: IInventory): Item[] {}

}

// Настройки бустерпака удачи

interface ILuckBoosterSettings extends IBoosterSettings {}

// Класс бустерпака удачи

class LuckBooster extends Booster {

constructor(settings: ILuckBoosterSettings) {}

}

// коллекция экземпляров бустерпаков, ключ - ID бустерпака

let boostersBase = {

1: new Booster({rarity: RARITY.RARE}), // пример добавления экземпляра бустерпака };

// API

// функция открытия бустерпака

function getBoosterLoot(boosterID: number, playerInventory: IInventory): Item[] { return boosters[boosterID].getBoosterLoot(playerInventory);

}

// TEST