Projekt: Dungeon Explorer

Programmiere ein textbasiertes Rollenspiel, in dem der Spieler mit einem Charakter durch einen Dungeon navigiert, gegen Monster kämpft und Schätze sammelt. Das Spiel soll verschiedene Charakterklassen, Waffen und Gegner enthalten.

**Technische Anforderungen:**

**1. Charaktere**

* Erstelle eine **abstrakte Klasse Character** mit folgenden Attributen:
  + String name
  + int health
  + int attackPower
  + **Methoden:**
    - abstract void attack(Character target)
    - boolean isAlive()
    - void takeDamage(int damage): Verringert health und gibt eine Meldung aus.
* Erstelle eine **Klasse Player**, die von Character erbt:
  + Enthält eine zusätzliche Item weapon, die die Angriffskraft erhöhen kann.
  + **Konstruktoren:**
    - Ein Konstruktor nimmt nur den name an und setzt Standardwerte.
    - Ein weiterer nimmt zusätzlich health und attackPower.
  + void equip(Item weapon): Der Spieler kann Waffen ausrüsten. Was passiert mit der AttackPower ?
  + void attack(Character target): Angriffslogik mit Waffe.
* Erstelle eine **Klasse Enemy**, die von Character erbt:
  + Implementiert attack(Character target).

**(1 und 2 in ca. 1h geschafft)**

**2. Waffen**

* Erstelle ein **Interface Item** mit Methoden:
  + String getName(): Gibt den Namen der Waffe zurück.
  + int getDamage(): Gibt den zusätzlichen Schaden zurück.
* Erstelle eine **Klasse Sword**, die Item implementiert: (
  + Enthält String name und int damage.
  + getName(), getDamage().

**3. Dungeon (das schwerste von der Aufgabe) (Erweiterung, „Teiloptional“)**

* Implementiere eine **Klasse DungeonGame**, die das Spiellogik enthält:
  + **Attribute:**
    - char[][] dungeon als 2D-Array (E für Gegner, C für Schätze, X für Ausgang).  
      Kann fest implementiert werden, oder wer mag mit Random-Funktion.
    - Weitere hängen von der Implementierung ab.
  + **Methoden:**
    - void start(): Startet das Spiel, steuert den Spielablauf.
    - void printDungeon(): Gibt das Dungeon mit der Spielerposition aus.
    - void movePlayer(char direction): Bewegt den Spieler (W, A, S, D).
    - void fight(): Lässt den Spieler gegen einen Gegner kämpfen.
    - void collectItem(): Lässt den Spieler ein Sword finden und ausrüsten
* BONUS:
  + Hinzufügen weiterer Player (Warrior, Mage,…).
    - Ermöglichen, dass ein Player am Anfang gewählt wird
    - Instanceof dadurch nutzen
  + Hinzufügen weiterer Items (Potions, Schild, Schätze, …)
    - Je nachdem müssen auch die Attribute oder Methoden der Klasse Player angepasst/erweitert werden
  + Umzingeln des Ausgangs mit mehr Feinden
  + Feinde in Rot markieren