Teamkampf

Einleitung

Teamkampf ist ein Simulator mit dem man rundenbasiert beliebig viele Teams gegeneinander antreten lassen kann. Pro Runde greift ein Mitglied eines Teams ein Mitglied eines anderen Teams an. Das angreifende Team wechselt pro Runde der Reihe nach durch (bei 3 Teams erst Team 1, dann Team 2, dann Team 3 dann wieder Team 1). Das angreifende Mitglied, angegriffene Team und angegriffene Mitglied, innerhalb des angegriffenen Teams, werden zufällig ermittelt.

Spiellogik

Es gibt in dem Spiel verschiedene „Wesen“ mit verschiedenen Stärken und Fähigkeiten. Sie sind momentan in zwei Hauptklassen: Monster -> Kobolde und Menschen -> Ritter unterteilt. Jedes „Wesen“ hat ein „Lagerplatz“ Attribut, z.b. 5. Jedes Team hat eine Maximallagerkapazität z.B. 50, das heißt das ein Wesen mit Lagerplatz = 5 maximal 10 mal in einem Team vorkommen kann und dieses Team dann voll ist. Es ist natürlich möglich unterhalb der Maximallagerkapazität zu bleiben und das Spiel zu starten.
Das Spiel läuft so lange bis nur noch ein Team lebendige Mitglieder hat. Dieses Team hat dann gewonnen.

Aufgabe: -> Code verstehen und Methoden anwenden!

Aus der Musterlösung wurden einige Codeausschnitte entfernt! Sie sollen die fehlende Logik implementieren.
Stellen an denen etwas entfernt wurde können Sie mit /* ? „Tipp“ */ erkennen.
Aber Vorsicht! Es könnte auch sein, das etwas ohne Markierung fehlt.
Sie müssen nicht exakt die Musterlösung erreichen. Die Logik / Funktion muss aber passen.

Bestender Code darf **NICHT** geändert werden.

Meist sind die fehlenden Abschnitte durch wenige Zeilen ersetzbar. Es ist aber möglich, dass Sie Helfer-Methoden implementieren müssen / können.

Nehmen Sie sich 30-60min Zeit den Code zu verstehen. Gehen Sie dafür im Kopf durch was ab der main Methode passiert.
Tipp: Sie müssen den Code nicht bis er terminiert durchgehen. Ab dem Moment wo die Schleife beginnt, welche die Runden zählt, passiert in jeder Runde das Selbe bis zum Ende.

Hinweise welche Konzepte angewandt / Aufgaben erledigt werden müssen:

-Verständnis der allgemeinen Spiellogik (Logik selbst implementieren)
-Comparable implementieren (Wesen müssen sortiert werden können)
-Lambdas / Funktionales-Interface „SpezialEffekt“ implementieren (*Ritter reduzieren Gegner Rüstung permanent um 50%, Kobolde verdoppeln ihre derzeitigen Leben*)
-Vererbung / Polymorphe (Alle Vererbungen und Implementierungen von Interfaces wurden entfernt)
-Singleton / Factory Klasse für Wesen (Methoden der Klasse „Wesenfactory“ ergänzen und die Klasse zu einem Singleton machen)
-Exceptions („WesenExistiertNichtException“ Klasse erstellen und an den korrekten Stellen einfügen)
-Abdeckung fehlerhafte Nutzereingaben (überprüfen was Nutzer eingibt. Nutzer ggf. erneut fragen -> Programm terminiert nicht)

Implementieren Sie zwei weitere Wesen um die Wesendiversität interessanter zu machen!
Fügen Sie dazu die Klassen ein und ergänzen Sie an den betreffenden Stellen die Logik um ihre Wesen im Spiel einzubinden.