

Football Match

Einleitung

Verwalten Sie die Ereignisse eines (Fußball)-Matches und stellen sie diese chronologisch dar.

Während eines Fußballmatches treten gewisse Spielereignisse (*MatchEvent*) auf, die in der Matchhistorie chronologisch angeführt werden, etwa Eckbälle oder Tore.

Matchtime

Jedes Spielereignis passiert zu einem bestimmten Zeitpunkt im Match (*MatchTime*).

Ein Match kann in folgende Segmente eingeteilt werden:

- Hälfte 1 (+Nachspielzeit) → *HALF_ONE*
- Hälfte 2 (+Nachspielzeit) → *HALF_TWO*
- Verlängerung 1 → *EXTRATIME_ONE*
- Verlängerung 2 → *EXTRATIME_TWO*
- Elfmeterschießen → *PENALTY_SHOOTOUT*

Die *MatchTime* ist aufgebaut aus einer Basiszeit *basetime* und der Nachspielzeit *overtime*. Alle Segmente - außer das Elfmeterschießen - können eine Nachspielzeit haben. Die erste Spielminute der Nachspielzeit der ersten Hälfte ist beispielsweise Minute 45+1, während die erste Spielminute der zweiten Hälfte Minute 46 ist. Beim Elfmeterschießen läuft die Zeit nicht mehr, es wird lediglich das Segment angegeben. Die natürliche Ordnung der *Matchtime* ist chronologisch, wobei Minute 45+1 **vor** Minute 46 liegt.

MatchEvent

Ein allgemeiner *MatchEvent* passiert zu einem bestimmten Zeitpunkt im Match, betrifft ein bestimmtes Team (Heimmannschaft *HOME* oder Gastmannschaft *GUEST*) und einen bestimmten Spieler (dargestellt als String).

MatchEvents haben eine textuelle Darstellung, z. B:

```
[22] Home: Goal by Alaba (assist Arnautovic)
[45+2] Guest: Goal by Holzhauser OG
[45+2] Home: Goal by Arnautovic P
[45+4] Guest: Goal by Ronaldo OG
[75] Guest: Goal by Arnautovic (assist Alaba)
[90+5] Guest: Goal by Arnautovic (assist Alaba)
[110] Guest: Corner by Kainz
[PEN] Home: Goal by Schöpf P
```

Die natürliche Ordnung der *MatchEvents* ist chronologisch.

In diesem Beispiel sollen zwei konkrete *MatchEvents* erstellt werden

- Eckbälle → Corner
- Tore → Goal

Bei Toren wird folgende Zusatzinformation miterfasst:

- Assist-Geber
- Strafstoß → *PENALTY*
- Eigentor → *OWNGOAL*

Main

Geben Sie die Demodaten auf drei Arten aus:

1. chronologisch sortiert,
2. nur Ereignisse der Gastmannschaft,
3. nur Ereignisse der Heimmannschaft.

```
[0] Guest: Goal by <Illegal MatchTime> (assist NN)
[20] Home: Goal by <Illegal Team> (assist NN)
[22] Home: Goal by Alaba (assist Arnautovic)
[45+2] Guest: Goal by Holzhauser OG
[45+2] Home: Goal by Arnautovic P
[45+4] Guest: Goal by Ronaldo OG
[75] Guest: Goal by Arnautovic (assist Alaba)
[90+5] Guest: Goal by Arnautovic (assist Alaba)
[110] Guest: Corner by Kainz
[PEN] Home: Goal by Schöpf P
```

GUEST:

```
[0] Guest: Goal by <Illegal MatchTime> (assist NN)
[45+2] Guest: Goal by Holzhauser OG
[45+4] Guest: Goal by Ronaldo OG
[75] Guest: Goal by Arnautovic (assist Alaba)
[90+5] Guest: Goal by Arnautovic (assist Alaba)
[110] Guest: Corner by Kainz
```

HOME:

```
[20] Home: Goal by <Illegal Team> (assist NN)
[22] Home: Goal by Alaba (assist Arnautovic)
[45+2] Home: Goal by Arnautovic P
[PEN] Home: Goal by Schöpf P
```

Erzeugung der Demodaten:

```
List<MatchEvent> history = new LinkedList<MatchEvent>();
history.add(new Corner(Team.GUEST, new MatchTime(110), "Kainz"));
history.add(new Goal(Team.GUEST, new MatchTime(75), "Arnautovic", "Alaba"));
history.add(new Goal(Team.GUEST, new MatchTime(45, 4), "Ronaldo").ownGoal());
history.add(new Goal(Team.GUEST, new MatchTime(45, 2), "Holzhauser").ownGoal());
history.add(new Goal(Team.HOME, new MatchTime(150), "Schaub").penalty());
history.add(new Goal(Team.HOME, new MatchTime(45, 2), "Arnautovic").penalty());
history.add(new Goal(Team.GUEST, new MatchTime(90,5), "Arnautovic", "Alaba"));
history.add(new Goal(Team.HOME, new MatchTime(22), "Alaba", "Arnautovic"));
history.add(new Goal(Team.GUEST, new MatchTime(-9), "<Illegal MatchTime>", "NN"));
```

Aufgabe

Implementieren Sie die Klassen *MatchEvent*, *Goal*, *Corner*, sowie *MatchTime* und *Main* laut Spezifikation.

