Naam Student: Stefan Klok  
Datum: 7-11-2023  
Opdrachtnummer: 7 Voetbalclub

# Probleemdefinitie

## Opdracht

FC Emmen, de lokale voetbalclub van Emmen wil het aantal overtredingen bij houden van de spelers in een team. Elk team heeft elf spelers. De club heeft meerdere teams. Ze willen het aantal gele, rode en zwarte kaarten bijhouden per speler. De speler betaald een boete voor elke kaart die hij krijgt. De boete voor een gele kaart is €18,32. Voor een rode kaart is dit €41,60 en voor zwart is dit €349,76. Met een zwarte kaart wordt je ook gediskwalificeerd en mag je niet meer meespelen in het team. FC Emmen wil weten hoeveel geld ze moeten betalen aan de National Soccer Association. De speler met het minst aantal kaarten wint de Fairplay Award. Dit moet worden geprint naar het scherm.

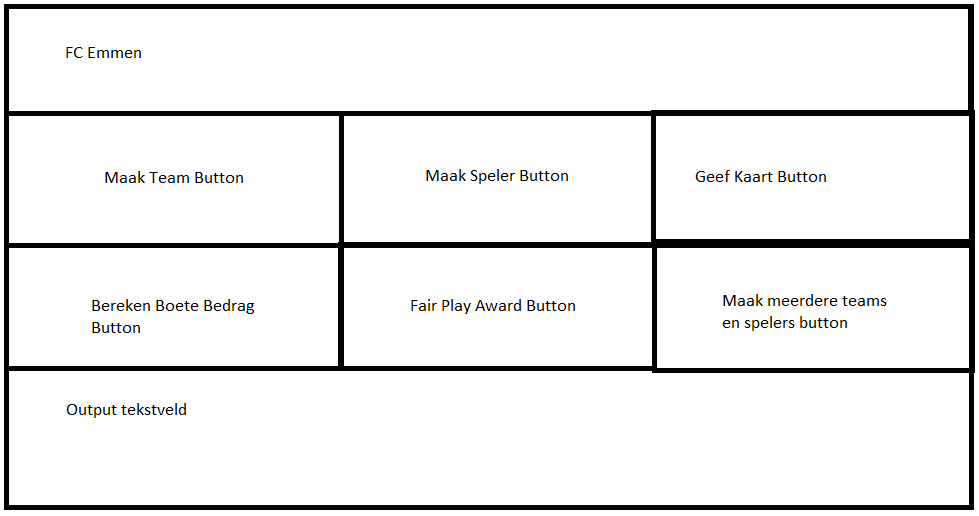
Samenvatting  
FC Emmen wil het aantal overtredingen bijhouden van spelers in een team.  
FC Emmen wil weten hoeveel geld ze moeten betalen aan de National Soccer Association.  
De speler met het minst aantal kaarten wint de Fairplay Award.

Invoer: Teams  
 Spelers (11 per team)  
 Kaarten

Uitvoer: Het bedrag dat FC Emmen aan de National Soccer Association moet betalen in euro’s  
 De speler met het minst aantal kaarten

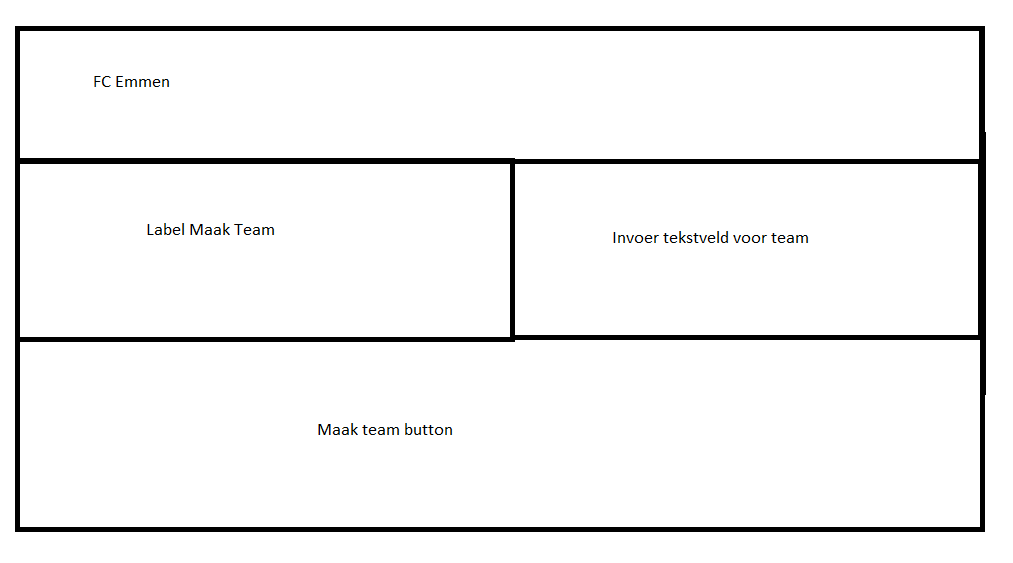
# Lay-out

Figuur: Het hoofdscherm met buttons voor acties en een tekstveld voor de uitvoer



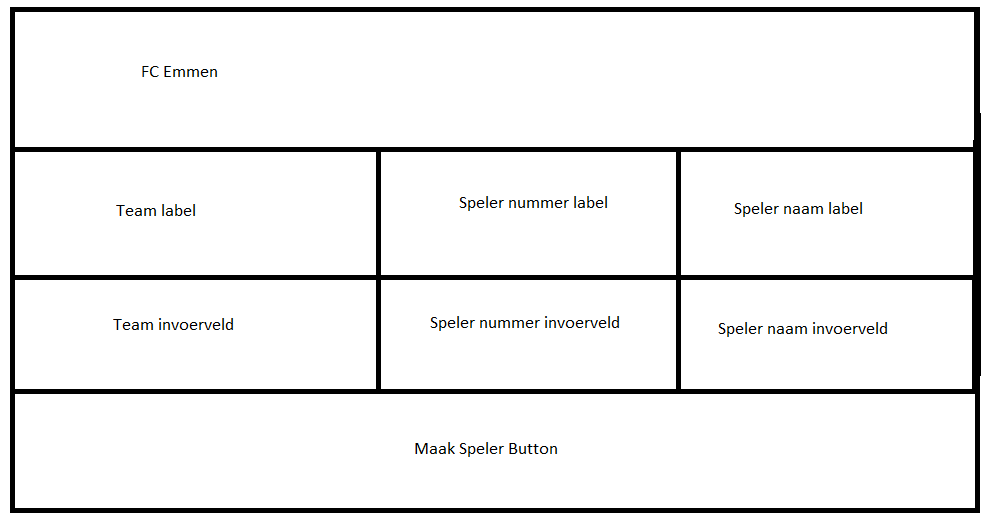
Figuur 1

Figuur 2: Het scherm waar een Team aangemaakt kan worden



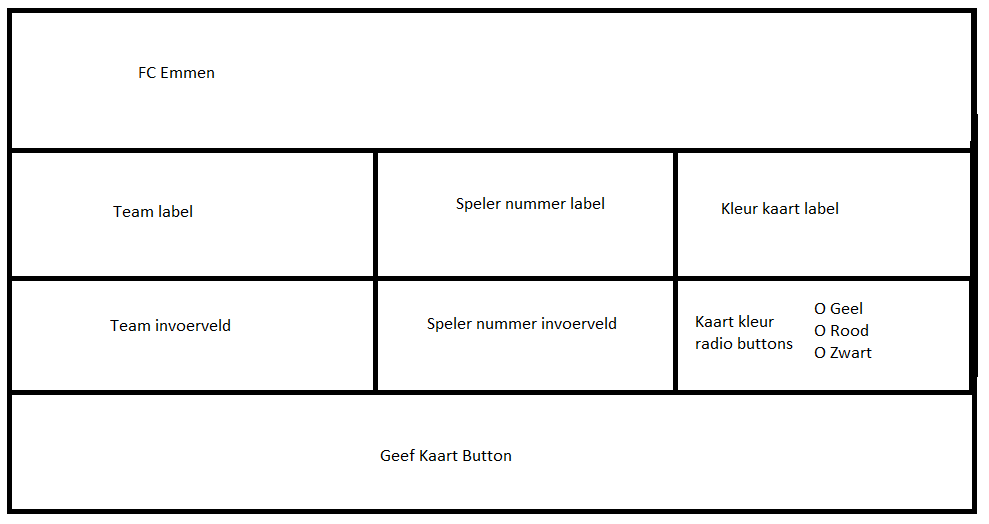
Figuur 2

Figuur 3 het scherm waar een Speler aangemaakt kan worden



Figuur 3

Figuur 4: Het scherm waar een kaart gegeven kan worden



Figuur 4

# UML

Afbeelding met tekst, diagram, Technische tekening, Plan

Automatisch gegenereerde beschrijving

Voor een duidelijkere afbeelding zie FCEmmenUML.jpg of het VioletUML bestand.

# Testgegevens

**Testgegevens voor de applicatie**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actie** | **Bestaande situatie** | **Invoer** | **Uitvoer** |
| Voeg een team toe |  | Team invoerveld: 1 | Output tekstveld: “Team is toegevoegd” |
| Voeg een speler toe | Team met nummer 1 is al gemaakt | Team invoerveld: 1  Speler nummer: 1  Speler naam: Stefan | Output tekstveld:  “Speler is toegevoegd” |
| Geef een kaart | Team met nummer 1  Speler met nummer 1 | Team invoerveld: 1  Speler nummer: 1  Kaart kleur radio: Geel | Output tekstveld: “Kaart is gegeven” |
| Voeg een bestaand team toe | Team met nummer 1 | Team invoerveld 1 | Team invoerveld: “Team bestaat al” |
| Voeg een bestaande speler toe | Team met nummer 1  Speler met nummer 1 | Team invoerveld 1  Speler nummer 1  Speler naam: Stefan | Invoer velden: “Speler bestaat al” |
| Voeg een Team met teamnummer a toe |  | Team invoerveld: a | Team invoerveld: “Voer hier een getal in” |
| Voeg een Speler met rugnummer x toe | Team met nummer 1 | Team invoerveld: 1  Speler nummer: x  Speler naam: Stefan | Team invoerveld: “Voer hier een getal in”  Speler nummer invoerveld: “Voer hier een getal in |
| Selecteer geen kleur bij het geven van een kaart | Team met nummer 1  Speler met nummer 1 | Team invoerveld: 1  Speler nummer: 1  Kaart kleur radio: geen | Kaart kleur label: “Kaart kleur: Selecteer een kleur” |
| Klik op de knop “Maak alles aan” |  |  | Output tekstveld: “Er zijn 2 teams en 22 spelers aangemaakt en er zijn kaarten gegeven” |
| Klik op de knop “Maak alles aan” wanneer er al een team bestaat | Team met nummer 1 | Klik op knop “Maak alles aan” | Output tekstveld: “Er bestaan al teams en/of spelers” |
| Klik op de knop “Bereken Boete” |  | Klik op knop “Bereken Boete” | Output tekstveld: “Boete bedrag: 0.0” |
| Klik op knop bereken boete wanneer er 1 team, 1 speler met een gele en zwarte kaart en 1 speler met een rode kaart is | Team met nummer 1  Speler met rugnummer 1 en naam “A”  Speler met rugnummer 2 en naam “B”  Een gele en zwarte kaart voor Speler A  Een rode kaart voor Speler B | Klik op knop “Bereken Boete” | Output tekstveld: “Boete bedrag: 409.68” |
| Klik op de knop Fair Play Award |  | Klik op knop “Fair Play Award” | Output tekst veld: “Winnaar(s) Fair Play Award: ” |
| Klik op de knop “Fair Play Award” wanneer 1 team is, 1 speler met een gele kaart, 1 speler met een gele en zwarte kaart en 1 speler met een rode kaart | Team met nummer 1  Speler met rugnummer 1 en naam “A”  Speler met rugnummer 2 en naam “B”  Speler met rugnummer 3 en naam “C”  Een gele kaart voor speler A  Een gele en zwarte kaart voor speler B  Een rode kaart voor Speler C | Klik op knop “Fair Play Award” | Output tekstveld: “Winnaar(s) Fair Play Award: A” |

**Testgegevens voor FCEmmen class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actie** | **Invoer** | **(Verwachte uitvoer)** | **Hoe te testen** |
| Voeg nieuw team toe | Team met nummer 1 | Team met nummer 1 wordt aangemaakt | Team 1 bestaat  checkVoorTeam(1) == true |
| Voeg bestaande team toe | Team met nummer 1  Team met nummer 1 | Er wordt eerst 1 team aangemaakt en het volgende team wordt niet aangemaakt. | Het aantal teams = 1.  getTeams.size() == 1 |
| Voeg speler toe aan team | Team met nummer 1  Speler met teamnummer 1, rugnummer 1 en naam Stefan | De speler wordt aan team 1 toegevoegd. | Speler in team 1 met nummer 1 bestaat  checkVoorSpeler(1, 1) == true |
| Voeg bestaande speler toe | Team met nummer 1  Speler met teamnummer 1, rugnummer 1 en naam Stefan  Speler met teamnummer 1, rugnummer 1 en naam Stefan | Er wordt een speler met nummer 1 en naam Stefan aan team 1 toegevoegd maar de tweede speler met nummer 1 en naam Stefan wordt niet toegevoegd | Het aantal spelers in team 1 = 2  getTeam(1).getAlleSpelers().size() == 1 |
| Geef een gele kaart | Team met nummer 1  Speler met teamnummer 1, rugnummer 1 en naam Stefan  Een gele kaart (kaartKleur 1) | Er wordt een gele kaart aan de kaarten van speler 1 toegevoegd | Het aantal gele kaarten van speler 1 is 1  getTeams().get(teamnummer)  .getSpeler(rugnummer)  .getAantalKleurKaarten(kaartkleur) == 1 |
| Geef een rode kaart | Team met nummer 1  Speler met teamnummer 1, rugnummer 1 en naam Stefan  Een rode kaart (kaartKleur 2) | Er wordt een rode kaart aan de kaarten van speler 1 toegevoegd | Het aantal rode kaarten van speler 1 is 1  getTeams().get(teamnummer)  .getSpeler(rugnummer)  .getAantalKleurKaarten(kaartkleur) == 1 |
| Geef een zwarte kaart | Team met nummer 1  Speler met teamnummer 1, rugnummer 1 en naam Stefan  Een zwarte kaart (kaartKleur 3) | Er wordt een zwarte kaart aan de kaarten van speler 1 toegevoegd | Het aantal zwarte kaarten van speler 1 is 1  getTeams().get(teamnummer)  .getSpeler(rugnummer)  .getAantalKleurKaarten(kaartkleur) == 1 |
| Bepaal de speler met het minste aantal en de beste kaarten | Team met nummer 1  Speler met rugnummer 1 en naam “A”  Speler met rugnummer 2 en naam “B”  Speler met rugnummer 3 en naam “C”  Een gele kaart voor speler A  Een gele en zwarte kaart voor speler B  Een rode kaart voor Speler C | Speler A is de speler met het minste en de beste kaarten | getSpelersMinsteKaarten() == “A “ |
| Bepaal het boetebedrag van FC Emmen | Team met nummer 1  Speler met rugnummer 1 en naam “A”  Speler met rugnummer 2 en naam “B”  Een gele en zwarte kaart voor Speler A  Een rode kaart voor Speler B | Het boete bedrag is (18,32 + 41,60 + 349,76 = €409,68 | Het boete bedrag is (18,32 + 41,60 + 349,76 = 409,68  getBoeteBedrag() == 409.68 |

# Globale uitwerking van het programma

De volgende klassen zijn in het programma te vinden

|  |
| --- |
| **Main** |

|  |
| --- |
| **FCEmmen** |
| - teams : HashMap<Integer, Team> |
| + FCEmmen()  + getTeam(teamnummer : int) : Team  + addNewTeam(teamnumer : int) : void  + addSpelerToTeam(teamnummer : int, rugnummer : int, naam : String) : void  + geefKaart(teamnummer : int, rugnummer : int, kleur : Kleur) : void  + checkVoorTeam(teamnummer : int) : boolean  + checkVoorTeams() : boolean  + checkVoorSpeler(teamnummer : int, rugnummer : int) : boolean  + checkVoorSpelers() : boolean  + get Teams() : HashMap<Integer, Team>  + getBoeteBedrag() : double  + getSpelersMinsteKaarten() : String  - getMinsteAantalKaarten() : int  - getBesteKaarten() : int[]  - setAantalBesteKaarten() : int  - verwijderTeamKaarten() : HashMap<Integer, int[]> |

|  |
| --- |
| **Team** |
| - teamnummer : int  - spelers : HashMap<Integer, Speler> |
| + Team()  + setTeamnummer(teamnummer : int) : void  + getTeamnummer() : int  + addSpeler(naam : String, rugnummer : int) : void  + getSpeler(rugnummer : int) : Speler  + getAlleSpelers() : HashMap<Integer, Speler>  + geefKaart(rugnummer : int, kleur : Kleur) : void  + getBoeteTeam() : double  + getSpelerMetMinstKaarten(kaarten : int[]) : String  + getMinsteAantalKaarten() : int  + getBesteKaarten(kaarten : int) : int[]  - getMinsteKleurKaarten(spelers : HashMap<Integer, Speler>, kleur : int) : int  - checkVoorSpelers() : boolean  - getEersteSpeler() : Speler |

|  |
| --- |
| **Enum Kleur** |

|  |
| --- |
| **Speler** |
| - naam : String  - rugnummer : int  - gediskwalificeerd : boolean  - kaarten : HashMap<Integer, Kaart> |
| + Speler()  + getNaam() : String  + setNaam(naam : String) : void  + getRugnummer() : int  + setRugnummer(int : rugnummer) : void  + setGediskwalificeerd(gediskwalificeerd : boolean) : void  + getBoete() : double  + getAantalKleurKaarten(kleur : int) : int  + geefKaart(kleur : int) : void  + getAantalKaarten() : int |

|  |
| --- |
| **Kaart** |
| # boete : double  # kleur : Kleur |
| + Kaart()  + setKleur(kleur : Kleur) : void  + getKleur() : Kleur  + setBoete(boete : double) : void  + getBoete() : double |

|  |
| --- |
| **GeleKaart** |
|  |
| + GeleKaart() |

|  |
| --- |
| **RodeKaart** |
|  |
| + RodeKaart() |

|  |
| --- |
| **ZwarteKaart** |
|  |
| + ZwarteKaart() |