Inlämningsuppgift 3

Grupp: 12

Simon Olofsson (siol0547)

|  |
| --- |
| Objektorienterad programmering  Höstterminen 2015  Kursansvarig: Henrik Bergström |

# Design

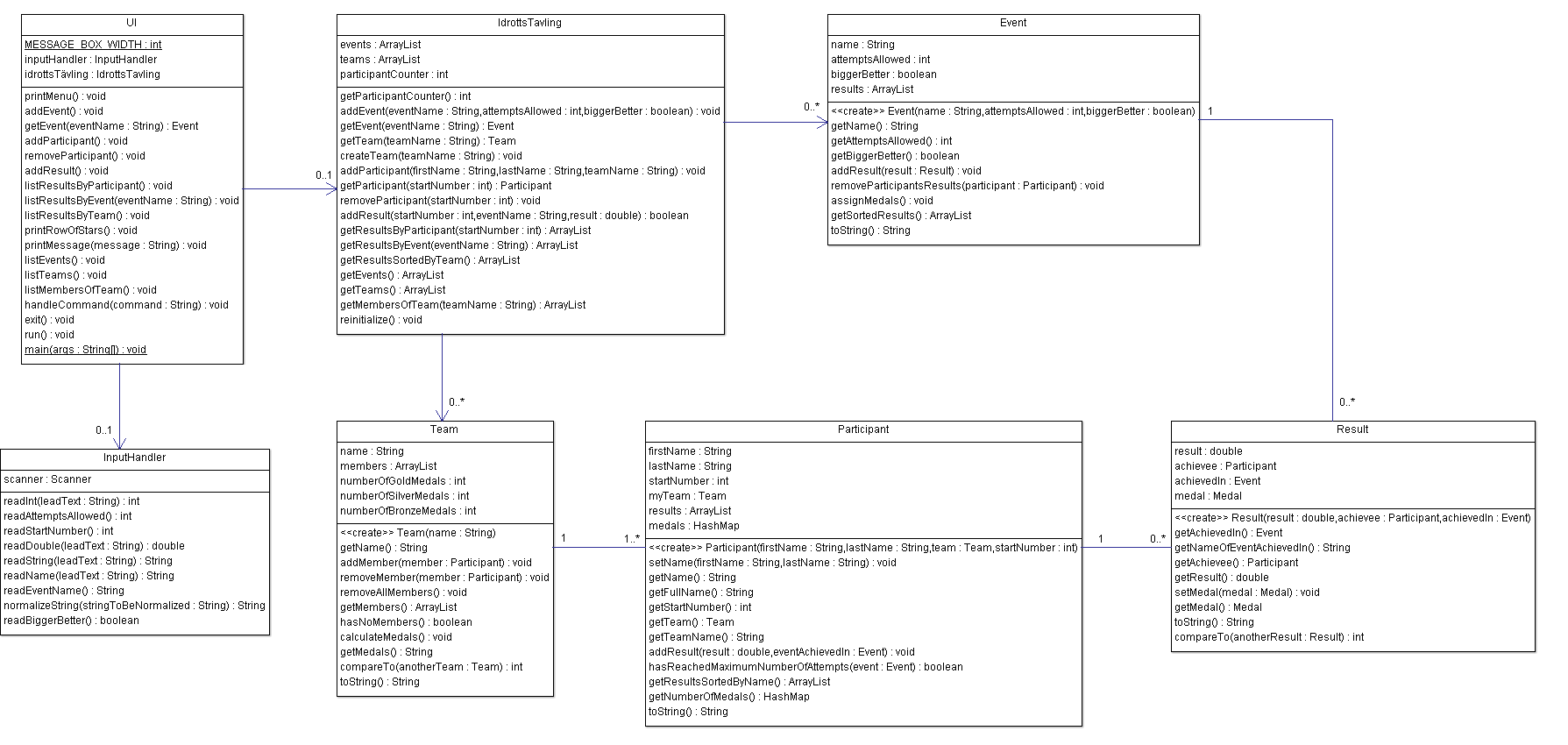


Bild 1. Klassdiagram.

Den övergripande designen är framförallt ett resultat av ett försök att upprätthålla så hög cohesion och så låg coupling som möjligt. För det första finns en särskild UI-klass som endast sysslar med att ta in rådata och skicka till en controller, vars enda uppgift är att koordinera arbetet mellan de resterande klasserna och sedan returnera rådata till UI-klassen. Som ett resultat av detta bör det inte vara några problem att lägga till ett grafiskt användargränssnitt utan några större ändringar än att gränssnittet kodas och kopplas till controllern.

Controllern har bara direkt tillgång till klasserna Team och Event, eftersom

# Funktionen lägg till resultat

Här ska ni beskriva funktionen *lägg till resultat* på ett sådant sätt att man tydligt kan följa flödet. Det är inte ett krav att göra så, men ett aktivitet- och/eller sekvensdiagram kompletterat med text och utdrag ur koden är antagligen ett lämpligt format.

# Funktonen resultatlista för lagen

Samma som ovan, fast för funktionen *resultatlista för lagen*.

# Normalisering av namn

Enligt kraven ska ni tänka på att undvika kodupprepning, och speciellt nämndes normaliseringen av namn. Här ska ni därför ge en kort beskrivning av hur ni gått tillväga för att undvika kodupprepning i just detta fall.

# Arrayer och ArrayList

Ett annat krav var att ni skulle använda minst en array och minst en ArrayList (eller annan datasamlingsklass) i programmet. Här ska ni därför ge en kort beskrivning av var ni gjorde detta och varför ni valde att använda dem på just dessa ställen.

Om det redan framgått från beskrivningarna ovan hur ni har använt arrayer respektive ArrayList så kan ni nöja er med att bara hänvisa dit.

# Statiska variabler och metoder

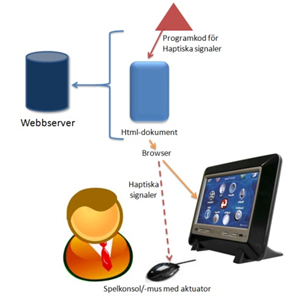
Om ni använt några statiska variabler eller metoder förutom main-metoden ska dessa motiveras här. Om ni inte gjort det så räcker det med att säga det. Konstanter räcker normalt med att redovisa, de behöver bara motiveras om deras användning inte är klar.

# Reflektion

Den sista delen i rapporten är avsedd för egenreflektion. Vad har fungerat bra? Vad har fungerat dåligt? Var uppgiften för lätt? För svår? Hur har ni tänkt när det gäller designen, namngivningen, skyddsnivåer, etc. Är det något ni borde ha övat mer på innan ni satte igång? Är det något kursen borde ha ägnat mer tid åt eftersom ni behövde det? Som vanligt finns det inga speciella krav på omfattning, men ½-1 sida bör vara ganska lagom.

# Mallanvändningen

Detta avsnitt tar upp några vanliga frågor/problem med mallandvändningen. Det ska naturligtvis plockas bort i den version ni lämnar in.



1. Systemillustration

## Bilder

Bilder ska tillföra något till texten, vara tydliga, ha en förklarande bildtext, och sitta på ett vettigt ställe i förhållande till texten. De bör också refereras till i texten.

### Rubriknivå 3

Rubriknivå tre och neråt bör man normalt undvika att använda. Ibland går det inte, men ofta är det ett tecken på att man delar upp sitt dokument i för små bitar.

## Referenser

De senaste Word-versionerna har ett referenshanteringssystem inbyggt. Det är lite rudimentärt, men fungerar hyggligt, och om man inte har tillgång till något bättre så bör man naturligtvis använda det. På OOP använder vi IEEE:s format med numrerade referenser inom hakparanteser. En referens till kursboken ska alltså se ut så här: [1], eller så här om man vill ha med sidanvisningar: [1, p. 4].

Om ni inte refererar till några källor ska ni plocka bort källförteckningen.

## Kod

Kod i dokumentet ska vara läsbar och helst formaterad som i en utvecklingsmiljö. Ett tips är att inte använda skärmdumpar. De blir nästan alltid oläsliga, och det går inte att få ut källkoden från dem om man skulle vilja.

## Avsnitt i mallen

Den text som finns i mallen från början är avsedd bara för att illustrera användningen och ska plockas bort innan ni lämnar ni.

Källförteckning

|  |  |
| --- | --- |
| [1] | J. Lewis och W. Loftus, Java Software Solutions, sjunde upplagan red., Pearson Education, 2012. |