

Inlämningsuppgift 3: Modsonger och Durin u.p.a.

Felix Cederblad (fece4341)

Objektorienterad
programmering

Höstterminen 2014

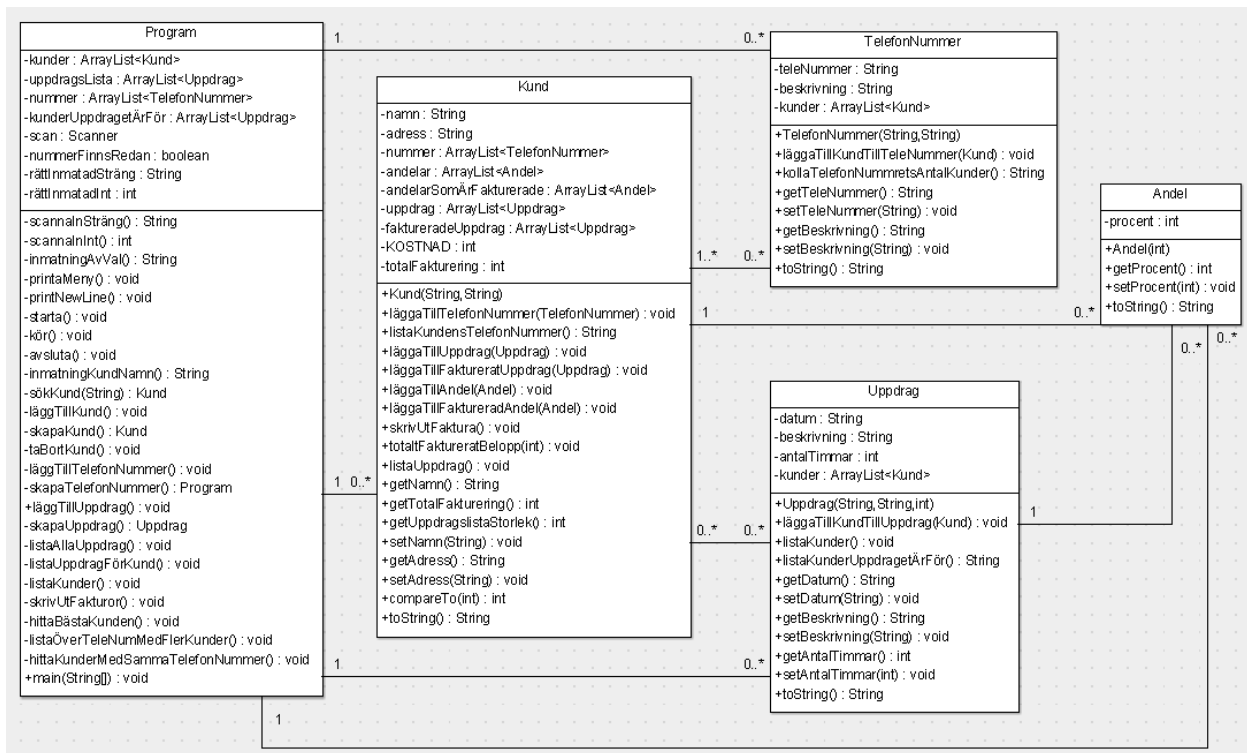
Kursansvarig: Henrik Bergström

1 Inledning

Vid skapandet av detta system var tanken att skapa ett system som klarade av både de funktionella kraven samt de icke funktionella. I början ritade jag lite på en skiss till ett klassdiagram som kunde illustrera ungefär hur de klasserna skulle kommunicera med varandra samt vilka klasser som behövdes. Vid det skedet var det dags att börja koda samt börja testa funktioner som kunde hjälpa till med att uppfylla kraven.

Under kodperioden krävdes det mycket ansträngning för att framställa rimliga lösningar som passade in i systemet så att inte andra funktioner blev drabbade. Införande av hjälpmetoder var något som jag utnyttjade vid passande tillfällen för att förenkla läsandet och förståelsen för koden. Systemet är nu byggt på ett sätt så att det erbjuder en relativt enkel och smidig textbaserad layout som erbjuder en meny av funktioner som användaren kan använda.

2 Design



(Variablerna och metoderna i Program ska vara understrukna då de är statiska men jag hittade ingenstans i ArgoUML att göra det).

Designen utgår ifrån att Programklassen är där mainmetoden finns och där programmet körs och att man även där ska kunna skapa objekt av klasserna Kund, TelefonNummer, Uppdrag och Andel och spara dem i en egen lista i klassen samt i en annan lista i en av de uppräknade klasserna beroende på vilka klasser som kommunicerar med vilka. Ett kundobjekt kan till exempel från Program klassens metoder få ett TelefonNummer objekt tilldelat sin egen TelefonNummer ArrayList. Färdiggjorda klasser som utnyttjas i detta system från Javas bibliotek är ArrayList[1], Collections[2] och Scanner[3]. Scannern utnyttjas via att en användare kan mata in värden på variabler, Collections genom Collections.sort(object o) för att sortera kundlistan enligt namn och sist ArrayList för att ha möjligheten till att spara obestämt antal objekt så att man inte behöver hårdkoda hur många platser antal objekt ska vara begränsade till.

3 Reflektion

Denna uppgift tyckte jag skiljde sig enormt från hundregistret. De var lika i sättet de var uppbyggda på men den stora skillnaden var att storleken på denna uppgift medförde många fler problem att ta hänsyn till. Vissa lösningar som skulle potentiellt sätt fungera för en funktion isolerad fungerade inte tillsammans med de andra lösningarna i uppgiften. Detta fick en att lägga mycket av tiden på problemlösning och ett försök till att implementera den kunskapen man har om java trots att den hittills är relativt begränsad.

Jag tycker att det är en bra uppgift men dock tycker jag att hoppet från hundregistret (med ett fåtal lektioner emellan) var ett stort hopp och tror att om jag inte hade tagit en programmeringskurs i gymnasiet att det hade varit svårt att få till en fungerande inlämningsuppgift samt en godkänd tenta.

Syntax inläringen tycker jag har varit bra, allt som krävdes i denna uppgift har vi fått lära oss. Däremot tycker jag personligen att lite mer betoning ska läggas på problemlösning och sådan tyst kunskap som är mycket svår att tänka ut själv om man är nybörjare inom området, men det är endast min åsikt.

Källförteckning

- [1] <http://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html> 18/1/ 20:06
- [2] <http://docs.oracle.com/javase/7/docs/api/java/util/Collections.html> 18/1/ 20:06
- [3] <http://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html> 18/1/ 20:06