

Inlämningsuppgift 2 - refaktorering, dokumentering

12-16 nov 2020

- Uppgift:** Använd kodbas Inlamning_2_ra_kod på
https://github.com/TomasKindahl/Inlamning_2_ra_kod
Ni skall inte tillföra ny funktionalitet, ni skall bara snygga upp koden!
- Tid:** 12-16 nov, inlämning senast 16 nov 12.30. Torsdag 12 nov eftermiddag och hela fredag bör räcka för att refaktorerar och dokumentera.
- Leverans:** *Favoritmetod:* Gör en egen kodbas och publicera på github, skicka länk till TomKi på slack.
En annan metod: Skicka Program.cs på slack till TomKi.

DEL 1 - refaktorering

Godkännitnivå, minst 12 p av följande

- Uppgift 1** (4 p) Refaktorerar så att adresslistan laddas i en egen static-metod.
- Uppgift 2** (12 p) Refaktorerar så att ingen else-if-gren

```
else if (command == "något")
```

har mer än fem rader kod i blocket – OBS: rader med bara { eller } räknas inte!
- Uppgift 3** (4 p) Skapa en metod **Person.Print** för utskrift av en **Person**, den skall anropas på ett ställe i resten av koden. Leta först reda på stället! Bestäm hur **Print** skall se ut! Skriv metoden, och anropa den från det andra stället i koden.

Väl-godkänt-nivå, minst 4 p av följande

- Uppgift 4** (4 p) Skapa en konstruktor i **Person** som inte tar argument, utan i stället frågar användaren efter namn, adress, telefon och email genom **Writes** och **ReadLines**! Använd denna nya konstruktor där koden skapar en ny användare genom att fråga användaren efter ovanstående information.
- Uppgift 5** (4 p) Skapa en metod i **Person**, med två parametrar:
a. den första tar emot en sträng som kan vara "**namn**", "**adress**", "**telefon**" eller "**email**", och
b. den andra innehåller en **string** som är ett namn-, adress- osv. värde
Denna metod sätter ett av attributen **namn**, **adress**, **telefon**, eller **email** i objektet beroende på vad det första argumentet var.

DEL 2 - kommentering

Godkännitnivå, minst 6 p av följande:

- Uppgift 6** (2 p) Gå igenom namn för "entiteter": variabler, metoder, klasser.
• Byt ut alla variabelnamn mot engelska
- Uppgift 7** (6 p) Skriv **begripliga** API-kommentarer till alla **static**-metoder, de skall ha formen:

```
/* METHOD: metodnamn (static)
 * PURPOSE: hur den används
 * PARAMETERS: alla parametrarnas namn och innebörd
 * RETURN VALUE: returvärdets innebörd
 */
```

Byt ut lutande text mot egna **begripliga** kommentarer på engelska.

Uppgift 8 (2 p) Skriv begripliga API-kommentarer till klassen **Person**, den skall ha formen:

```
/* CLASS: namn
 * PURPOSE: var den används
 */
```

Byt ut lutande text mot egna **begripliga** kommentarer på engelska.

OBS: fortsätter på nästa sida!

Uppgift 9 (2 p) Skriv en begriplig API-kommentarer till metoden **Print** i klassen **Person**.

Väl-godkänt-nivå, minst 2 p av följande

Uppgift 10 (X p) Skriv **begripliga** API-kommentarer till den/de **Person**-metoder, och den/de konstruktörer du skapade i uppgifterna 4 och/eller 5, två poäng per konstruktor/metod:

```
/* METHOD: metodnamn (static)
 * PURPOSE: hur den används
 * PARAMETERS: alla parametrarnas namn och innebörd
 * RETURN VALUE: returvärdets innebörd
 */
```

Byt ut lutande text mot egna **begripliga** kommentarer på engelska.