

# AAU, Programmering i Java

## Intern skriftlig prøve

**16. juni 2010**

Opgavebesvarelsen skal afleveres via email til [ulrik@cs.aau.dk](mailto:ulrik@cs.aau.dk) og [bst@tnb.aau.dk](mailto:bst@tnb.aau.dk)

Når opgaven afleveres skal der stå **PJAVA: studienummer** i emnefeltet.

Opgaven skal afleveres som en Java fil, der indeholder kildekoden. En vigtig del af opgaven består også i at skrive kommentarer til opgaveløsningerne. Det er valgfrit om man skriver sine kommentarer på dansk eller engelsk. Opgaven skal løses individuelt. I bedømmelsen af opgaven lægges vægt på at løsningen er korrekt og let læselig og ikke på hastigheden eller hukommelsesforbruget af koden.

Opgaven består af dette dokument, der beskriver opgaven samt en Java fil der hedder JavaExam2010.java, der skal redigeres og returneres som opgavebesvarelsen.

Lav noget kode i main metoden der tester alle de implementerede funktioner.

Det er i opgaven hverken nødvendigt eller tilladt at anvende TextIO.java fra kurset.

Den øverste kommentarblok i besvarelsen skal være forsynet med navn og studienummer.

Besvarelsen skal være afleveret senest den 18. juni 2010 kl 12:00.

Pensum:

Pensum beskrives i forhold til lærebogen: Introduction to Programming Using Java, Fifth Edition, David J. Eck.

Pensum er følgende afsnit i bogen:

Afsnit 1.1-1.5

Afsnit 2.1-2.6

Afsnit 3.1-3.7

Afsnit 4.1-4.5

Afsnit 5.1-5.7

Afsnit 7.1-7.5

Afsnit 9.1

## Opgave 1

Denne metode (`dist`) skal beregne afstanden imellem to punkter på en todimensionel flade. Input variablene `x1` og `y1` skal sammen fortolkes som et punkt og `x2` og `y2` skal sammen fortolkes som det andet punkt.

## Opgave 2

Denne metode tager en liste af strenge (`ArrayList<String>`) der muligvis indeholder dubletter. Med dubletter menes strenge, der indeholder identisk tekst og ikke referencer til det eksakt samme objekt. Når metoden finder to eller flere dubletter skal alle dubletterne ændres således at de får et tal tilføjet. Den første udgave af dubletten skal have et 1 tilføjet, den anden 2 osv. Den modificerede liste skal være tilgængelig for den, der har kaldt funktionen ved at den originale input liste modificeres. I den rigtige retur værdi skal det maksimale antal fundne identiske strenge returneres.

Eksempel: {"hund", "kat", "kat", "ged", "hund"} -> {"hund1", "kat1", "kat2", "ged", "hund2"}

## Opgave 3

Denne metode tager et array af doubles som input. Den skal returnerer gennemsnittet af disse, hvor den ser bort fra X største og mindste værdier. Hvor X er angivet af input variabelen `exclude`. Hvis `exclude` er 1 så skal den største og den mindste værdi fjernes. Hvis `exclude` er 2 så skal de to største og de to mindste værdier fjernes. Hvis `exclude` er 3 så skal de tre største og de tre mindste værdier fjernes og så fremdeles. Hvis dette resulterer i at der ikke er nogen værdier tilbage at regne gennemsnit af eller at der er for få værdier at fjerne skal metoden kaste en `IllegalArgumentException`.

Hint: Lav eventuelt en hjælpe metode, der fjerner den mindste og den største værdi og returnerer arrayet.

## Opgave 4

Denne opgave går ud på at lave et lille system, der kan holde styr på regningerne i en restaurant. Systemet skal ikke være interaktivt og derfor er det ikke nødvendigt eller tilladt at benytte `TextIO.java` fra kurset.

Kernen i systemet er klassen `Restaurant`, der er lavet som en helt tom klasse i bunden af filen `JavaExam2010.java`. På denne klasse skal der defineres en metode (`addTable`) der tilføjer et bord (med nummer) til restauranten. Ud over dette skal det være muligt at oprette en regning (`addBill`) der knytter sig til et bestemt bord. Det er valgfrit om man vil lave ekstra hjælpeklasser, der repræsenterer et bord og en regning.

På klassen `Restaurant` skal der også være en metode til at få den samlede omsætning for et givet bord, samt en metode til at få den samlede omsætning for hele restauranten.

Lav i metoden `Opgave4` noget kode der først opretter et `Restaurant` objekt og derefter tilføjer nogle borde og regninger til dette objekt. Til sidst skal systemet benytte de to sammentællings funktioner. Alt dette skal demonstreres således at det bliver printet ud på `System.out` når metoden `Opgave4` kaldes.