

Javaprogrammering 60YHP - HT23

Labb 1 + muntligt test

Deadline (Söndag) 2023-10-08, kl 23:59

Betygskriterier som täcks i denna examination:

- ☐ **G5** **kunskaper om...** Objektorienterad programmering.
- ☐ **G6** **kunskaper om...** Generics.
- ☐ **G9** **kunskaper om...** Versionshantering.
- ☐ **G12** **färdigheter i att...** Programmera med Java.
- ☐ **G13** **färdigheter i att...** Strukturera algoritmer och datorprogram.
- ☐ **G15** **färdigheter i att...** Skriva lättförståelig kod.
- ☐ **G17** **färdigheter i att...** Skapa feltoleranta applikationer och system.

Vad är **kunskaper**?

Den studerande har **kunskaper** (utgörs av fakta, principer, teorier och praxis som är kopplade till ett arbets- eller studieområde).

Vad är **färdigheter**?

Den studerande har **färdigheter** (är förmåga att tillämpa kunskaper och beprövad erfarenhet för att utföra uppgifter och lösa problem).

Innehåll

| | |
|------------------------------------------------------------------|---|
| Labb 1: Frukt- och Grönt-vågen..... | 4 |
| Del A – Konsolbaserade meny 5 | 5 |
| Del B – Skapa och ta bort en vara/produkt (class Product)..... 6 | 6 |
| Del C – Söka efter en produkt/vara..... 7 | 7 |
| Muntligt test: Frukt- och Grönt-vågen..... 8 | 8 |

Övrig information

- Denna uppgift ska **inte lösas med hjälp av arv** (inheritence).
- Står någonstans att du ska kunna **förklara/motivera** något så menas det att du kommer behöva göra detta i det muntliga testet (inspelningen) där du presenterar och redogör ditt program som du också lämnar in i samband med din kod.
- Se till att du är väl **införstådd i examinationens betygsriterier** för att säkerställa att du täcker av dessa genom ditt program och inspelning. T.ex. är G15 ett betygsriterie som täcks i denna examination vilket syftar på att du ska visa i din inlämning att du har färdigheter i att kunna skriva lättförståelig kod vilket du behöver tänka på att uppfylla innan du lämnar in ditt arbete.
- Det måste finnas och användas **minst en Array** och minst en **ArrayList** i programmet.
- Alla kod-filer som lämnas in behöver det stå ditt förnamn, efternamn och ITHS-mail.
- Labb 1 och muntliga testet är individuellt arbete. Därför är det viktigt att det är du själv som har gjort arbetet. Du får arbeta med andra personer men du ska själv ha skrivit koden, inte vara plagierat samt att du förstår och kan förklara din egen kod. Vid misstanke av plagierat arbete eller AI-genererat arbete så kan utbildare kalla dig till en muntlig redovisning på plats eller digitalt av det inlämnade arbetet och eventuellt ge en ny examinationsuppgift som täcker samma betygsriterier.
- Läs hela detta dokument innan du påbörjar ditt arbete.



Utbildarens anteckningar

| namn | efternamn |
|-------------|------------------|
| | |

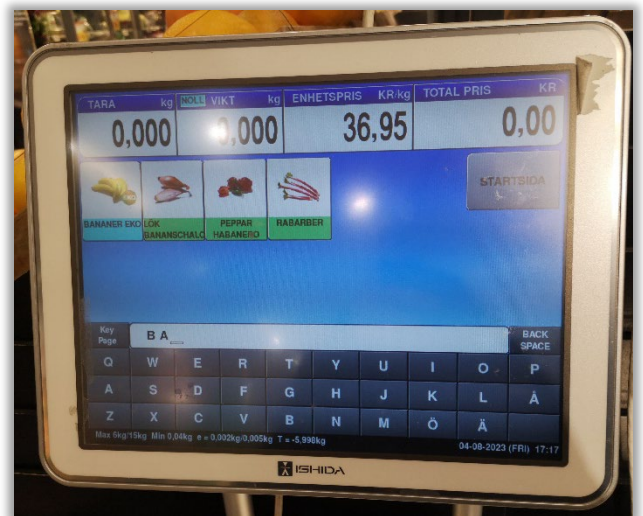
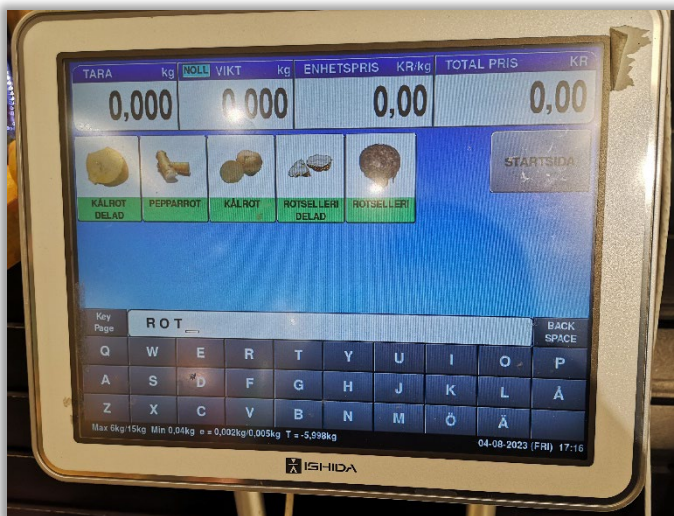
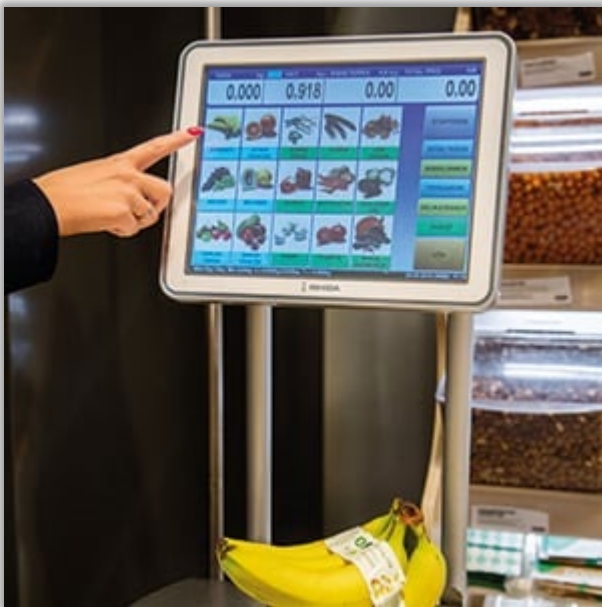
| Färdig / Komplettering | Kriterie | Kommentar |
|------------------------|------------|-----------|
| | G5 | |
| | G6 | |
| | G9 | |
| | G12 | |
| | G13 | |
| | G15 | |
| | G17 | |

Utbildarens anteckningar

Labb 1: Fukt- och Grönt-vågen

I matbutiker brukar det antingen vid kassorna eller i frukt- och grönt-avdelningen finnas en våg som beräknar priset på en specifik vara. Antingen är det styckpris eller viktpreis beroende på vad matbutiken själva har bestämt.

Din uppgift är att skapa ett program som beräknar priset på varje enskild vara utifrån om det är styckpris eller viktpreis, samt vilken vara det gäller då det kan vara specialerbjudanden (kampanjpriser) för specifika varor eller varugrupper.



Del A – Konsolbaserade menyn

Java-programmet ska vara konsolbaserat vilket innebär att programmet ska styras av användaren genom konsolen. **Användaren är den anställda i matbutiken som sköter om och administrerar själva Frukt- och grönt-vågen** (alltså inte en kund). I programmet behöver det:

- Tydligt framgå för användaren **vad som kan/ska skrivas in** för att göra specifika saker i programmet.
- Tydligt framgå att programmet har **startats och avslutats**.
- Kunna hantera när användaren **skriver in felaktigt utan att programmet kraschar** och därefter informera användaren vad som behöver/måste skrivas in.
- Programmet **får inte någonstans krascha pga. att användaren skriver in på felaktigt** sätt. Du behöver då testa ofta och be någon annan prova ditt program för att upptäcka eventuella problem.
- Använda **loopar för att köra programmet** så länge användaren vill det. Tänk på att du ska kunna motivera varför du väljer vissa loopar framför andra.
- Använda **villkorssatser för att styra** vad som ska hända beroende på vad användaren skriver in. Tänk på att du ska kunna motivera varför du väljer vissa villkorssatser framför andra.
- Att **inte ha redundant (upprepande) kod**. Om kod upprepas ska du kunna motivera varför det upprepas och kunna förklara vad du har försökt att göra för att undvika redundant kod.

Del B – Skapa och ta bort en vara/produkt (class Product)

Den som är ansvarig för frukt- och grönt-avdelningen i matbutiken måste kunna lägga till och ta bort en vara/produkt. Detta ska göras med hjälp av objektorienterad programmering där **klassen heter Product** (en fil i projektet som heter **Product.java**).

Denna klass ska:

- Ha **egenskaperna (properties)**: name, price och productGroup. Du kan även ha fler egenskaper som du anser är passande/nödvändiga för ditt program. Du ska då kunna motivera och förklara varför du anser att dessa behövs i ditt program.
- Ha **tre konstruktorer**:
 - o en konstruktor som skapar en produkt med **namn**.
 - o en konstruktor som skapar en produkt med **namn** och **pris**.
 - o en konstruktor som skapar en produkt med **namn**, **pris** och **varugrupp/-er**.
- Ha lämpliga **getters** som returnerar värdet på en egenskap.
- Ha lämpliga **setters** som sätter/uppdaterar värdet på en egenskap.
- **Uppdaterad toString**-metod för en lämplig utskrift på objektets egenskaper.

Del C – Söka efter en produkt/vara

Som anställd i matbutiken vill man vara säker på att det som säljs på avdelningen faktiskt finns inmatat i systemet och att priserna stämmer. Därför ska det finnas en möjlighet för användaren ett sätt i programmet att söka/hitta:

- Användaren ska kunna söka/hitta en produkt/vara genom att **skriva fullständiga namnet eller del av namnet**. Om det är flera produkter/varor som matchar/innehåller delnamnet så ska alla dessa skrivas ut. T.ex. om användaren skriver "rot" så ska det exempelvis skrivas ut "morot", "kålrot", "pepparrot", "rotselleri" osv.
- Användaren ska kunna **navigera sig fram** till en produkt/vara. T.ex. "frukter" → "stenfrukter" → "nektarin". Oavsett hur användaren har navigerat sig fram så ska det alltid finnas möjlighet att:
 - o Gå tillbaka till huvudmenyn.
 - o Istället söka på produktens/varans namn.
- **När en produkt/vara har hittats** av användaren så ska detta skrivas ut i konsolen:
 - o Fullständiga namnet på produkten/varan.
 - o Vilka kategorier/grupper som produkten här till t.ex. "nektarin" har kategorierna/grupperna "frukter" och "stenfrukter".
 - o Styckpris eller kilopris.
 - o **Möjlighet att skriva in antal/mängden** av den produkten som sedan skriver ut korrekt pris beroende på produktens/varans prissättning. T.ex. om bananer säljs för 19,50 kr/kg så ska användaren kunna skriva vikten 2,25kg vilket i detta fall skriver ut ett lämpligt meddelande följt av slutpriset med två decimalers noggrannhet (avrundning) 43,88kr (19,50kr/kg * 2,25kr).

Muntligt test: Fukt- och Grönt-vågen

Utöver att utveckla kod ska även göra en inspelning på minst 5 minuter där du redovisar och visar ditt program. Detta kallas för ett muntligt test just för att du testas på hur du presenterar kod samt att du använder och förklarar begreppet på korrekt sätt.

Nedan kommer en checklista på saker att tänka på i samband med inspelningen:

- Presentera dig kort och ifall du har följt instruktionerna helt och hållet eller om du har gjort dina egna varianter på lösningar, vad du har gjort för egna varianter samt motivering till varför du har gjort på ditt sätt.
- Inspelningen ska vara mellan 5 - 10 minuter lång.
- Både du och din kod ska synas på inspelningen.
- Du kan använda dig av stödord i form av PowerPoint/presentationsverktyg men inte läsa ett manus innantill.
- Du ska demonstrera/visa på inspelningen hur du testar ditt program och visar att den fungerar enligt uppgiftsbeskrivningarna.
- Om din kod inte fungerar så som du har tänkt dig så är det helt okej så länge du:
 - o Förklarar vad det är för problem som uppstår.
 - o Förklarar varför just det problemet uppstår.
 - o Förklarar hur det ska/kan lösas.
 - o PS: uppstår det ett problem så får du mer än gärna visa i inspelningen hur du löser/felsöker problemet.
- Den kod du presenterar är din egen kod och är samma kod som du har lämnat in / skickat in för rättning.
- Du får själv välja vilket verktyg du använder för inspelning. Dock ska filen vara en .mp4-fil. Förslag på inspelningsverktyg: Teams eller OBS-studio (båda är gratis).