

Kassakris!

COKA är i matvarubutik som är i stort behov av ett nytt kassasystem. Då deras nuvarande DOS baserade kassastystem är för dyrt att bygga vidare på samt att det tar väldigt mycket tid för alla extrapersonal att lära sig. Samt att de som kan de börjar dö ut ;)

De har börjat skissa på ett GUI och fått en av de sista utvecklarna att börja innan hen också dog ut ;)

Se den koden nedan:

```
package kassasystem;

import javax.swing.*;
import java.awt.*;

class Kassa extends JFrame{
    final String[] VAROR = {"Gurka", "Paprika", "Tomat",
        "Mjölk", "Yoghurt", "Grädde",
        "Bröd", "Müsli", "Kakor"};

    public Kassa(){
        super("Kassa");
        JPanel north = new JPanel();
        add(north, BorderLayout.NORTH);

        JButton nyKundKnapp = new JButton("Ny kund");
        north.add(nyKundKnapp);

        JPanel center = new JPanel();
        center.setLayout(new GridLayout(2,1));
        add(center, BorderLayout.CENTER);
        JPanel knappar = new JPanel();
        center.add(knappar);
        knappar.setLayout(new GridLayout(3,3,10,10));
        for(int i = 0; i < VAROR.length; i++){
            JButton b = new JButton(VAROR[i]);
            knappar.add(b);
        }
        JTextArea display = new JTextArea();
        center.add(new JScrollPane(display));

        JPanel south = new JPanel();
        add(south, BorderLayout.SOUTH);
        south.add(new JLabel("Antal: "));
        south.add(new JTextField(5));
        south.add(new JLabel("Totalbelopp = "));
        south.add(new JLabel("0"));

        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(300, 300);
        setVisible(true);
    }

    public static void main(String[] args){
        new Kassa();
    }
}
```

Uppgift 1:

De hade tänkt sig att börja lite fint med bara några varor men skulle vilja ha in

- Smör
- Knäckebröd
- Ost

På en till rad under den understa.



Uppgift 2:

De börjar visa gult för personer och vill kunna visa på att det händer saker om man trycker på knapparna (konstiga användare). Så de tänker att det fort borde gå att lägga till så att i terminalen (Sysout) kommer ut texten "Ny kund" när man trycker på den knappen.

Uppgift 3:

Tog inte många sekunder att lista ut att när man kör programet så kan man inte se vad som skrivs ut i terminalen. Så din nya uppgift är att få till så att det istället för att skrivas i terminalen kommer en popupruta som det står på "Ny användare" och sedan har en OK-knapp.

Uppgift 4:

Sommarbemaningen börjar visa sig och få seniora kvar på kontoret för COKA men är ett gäng glada studenter som får fortsätta kravställa istället. De kommer fram till att OK-knappen på ny användare är onödig så nästa uppgift blir att ändra till en JA-Nej popupruta med frågan "Skapa ny användare".

- Vid JA
 - o Visa en OK-popupruta som säger "Ops, du svarade JA"
- Vid nej
 - o Visa en Ok- popupruta som säger "Ops, du svarade Nej"

Uppgift 5:

Sommarjobbarna AKA studenterna verkar inte riktigt ha koll på sin domän. Men efter lite mejlväxlande så tar du motvilligt och implementerar deras nästa steg.

Se till att det samtidigt som man trycker på "Ny kund" skrivs "en kund skapad" i textfältet. Där det skrivs innan någon av popup-rutorn syns.

Uppgift 6:

På samma tema tycker de att detta var riktigt nice och tycker att man kan använda Totalbeloppet på ett lite spännande sätt. Eller som de säger "ett nytänkande och

framåtdrivande användningsområde för något i vanliga fall så tråkigt”.

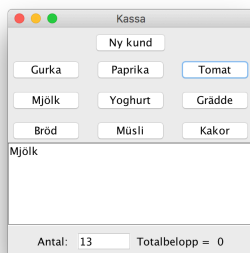
Så för varje gång man trycker på ”Ny kund” så se till att ”0:an ”ökar med ett så man kan se hur många gånger man tryckt på den knappen. Detta ska ske både innan popuprutorn och utskriften till textfältet.

Fråga 6:

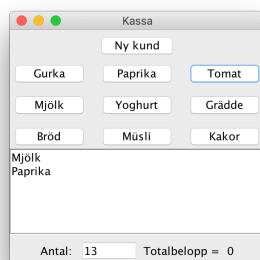
Efter en spännande vecka kommer en av de ordinarie från kravavdelningen tillbaka och funderar på vad sjutton som har hänt!

Efter att hen undrat vad som de faktiskt lagts tid på under gångna veckan så kommer nya direktiv. Så lägg ner allt arbete på konstiga ”Ny kund” som ju var det lägsta fokuset. Inte behöver man det när man demonstrerar en application.

Så nästa utgift blir istället att se till att när man trycker på någon av de matvarorna så kommer denna att läggas till i textfältet. Så om man trycker på ”Mjöl” första gången så står det:



I textrutan. Men om man sedan trycker på ”Paprika” så läggs det till så att det står:



(Notera att de exemplen ovan inte återspeglar hur det ska se ut mer än för just textfältet)

Uppgift 7:

Sommarjobbarna har fått nya uppgifter och det har bestämts att man ska se till att rensa bort lite av vad de gjorde.

Uppgiften blir att ta bort utskriften från textrutan när man trycker på ”Ny kund”.

Uppgift 8

Nästa steg är att de skulle vela ha att priset på varorna också skriv ut. Men de är inte allt för tekniskt kunnande så de ger dig i uppdrag att designa hur man kan koppla ett pris till en vara. Det visade sig att vikten även den är intressant så lägg till stöd för den också.

Din uppgift är att implementera det samtidigt som funktionaliteten som finns nu även ska fungera likadant efteråt. Ett till krav som ställs är att data inte ska sparas redundant. Så att det inte sparas mjölk eller annat på fler än ett ställe.

Uppgift 9:

Lite priser på de olika varorna är som följande

Namn	Pris	Vikt
Gurka	10	0.3
Paprik	7	0,2
Tomat	10	0,25
Mjölk	13	1
Yogurt	23	1
Grädde	14	0.3
Bröd	25	0,8
Müsli	30	1.1
Kakor	32	0.9
Smör	35	0.5
Knäckebröd	27	0.6
Ost	70	0.6

För tillfället gör det inget om de är hårdkodat utan lägg till alla värden så att de finns.

Uppgift 9:

Nu vill man dra nytta av att man även känner till pris så att det skrivs ut efter namnet på varan när man trycker på det.

Uppgift 10:

Man har även uppmärksammat att räknaren för hur många gånger man tryckt på Ny kund är kvar. Uppgiften är då att ta bort det så att den inte räknar upp.

Uppgift 11:

Istället vill man veta hur mycket det man handlar kostar.

Uppgiften blir då att man ska kunna fylla i ett värde i antal rutan och sedan trycka på en vara och då skrivs varan ut pricis som förut. Men att totalbeloppet ska uppdateras med så mycket varan är värd. Där man då får ta priset * antal och lägga på det på det föregående totalbeloppet.

I butiken kan man bara köra hela varor.

Uppgift 12:

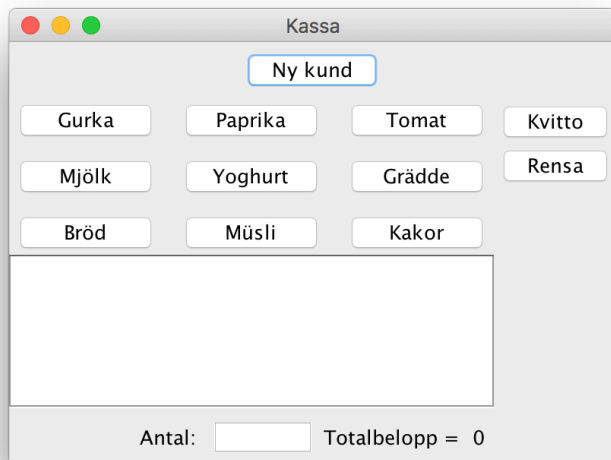
Studenternas nya uppgift de fick vara bland annat att testa och de får programmet bara att

Uppgiften är då att det inte ska crasha och inget ska läggas till så länge det inte är ett korrekt heltal.

Det känns som att det börjar bli ett fungerande kassa-system! Men kravställarna är lite osäker på vad som ska göras. De är överäns om att de behöver att prata med produktägaren. Men en kaffepaus blev första fokus för dem som blev lite "lång" så var lika bra att gå hem efter kaffepaus enligt deras utsaga. Men en sak du hört dem prata om var kvitto funktionalitet och pratat om de ska skrivas ut sorterat efter i första hand pris och i andra hand namn. Så du bestämmer dig för att förbereda och få upp din produktivitet. Så varor enkelt kan sorteras eller sparas i en trädstruktur.

Uppgift 14:

För kvittofunktionaliteten verkar vara något att satsa på men knappar för de saknas. Så nästa uppgift är att modifiera GUI:t så att det ser ut likt nedan:



Där även brädden har ändrats från 300 till 400.

Du pratade med en kompis lite innan och googlade och hittade lite spännande länkar till att läsa på under arbetet:

<https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

Uppgift 15:

Nu har det äntligen beslutats att kvitto är nästa sak som måste till!

Det har funderats på utseendet och det blir nu ditt uppdrag att implementera det.

Tydligen kan du arbeta i sömenen för du har detta att utgå från:

```
private JPanel kvitto(something something){
    JPanel kvitto = new JPanel();
    kvitto.setLayout(new BorderLayout());
    String kvittoText = "";

    //här skall logiken för att lägga till texten i kvittoTexten

    JTextArea kvittoTexten = new JTextArea(kvittoText);
    kvitto.add(new JScrollPane(kvittoTexten),BorderLayout.CENTER);
    return kvitto;
}
```

och

```
kvittoKnapp.addActionListener(x->{
    JFrame kvittoWinov = new JFrame("Kvitto");

    kvittoWinov.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    kvittoWinov.add(kvitto(somthing));
    kvittoWinov.setSize(150,350);
    kvittoWinov.setVisible(true);
});
```

så uppgiften blir att fylla i strängen kvittoText med alla varor och pris. Sen som på alla kvitton så ska det stå vilken butik det är från, tacka för besöket, totalpriset, varav hur mycket som är moms (räkna på 25%) är.

Uppgift 16:

Varorna är många och i de testerna som finns är det inte allt för många med. De varor som inte finns med i systemet får kassörerna skriva till på kvittot för hand. Vilket gör att många är sjukskrivna pga problem i med handleder... Så det måste till fler varor.

Men för att få till det måste de förberedas. För att kunna få till det så har du funderat ut att det behövs strukturella förändringar av hur varor sparas. Vilket du har berättat för teamet och fått som första uppgift. De kattergorierna är:

Mejerivaror	Grönsaker	bröd
Dryck	Shark	Torrvaror

Första uppgiften blir då att skapa en datasamling på sidan för att vidbehålla all funktionalitet som finns för GUI i dagsläget. För det får du bestämma hur de ska spara och implementeras på bästa sätt. Vad väntar du på, bara att sätta fart ;)

Uppgift 16b:

På natten efter drömde du inte om de hörliga kvällsdoppet eller goda middagen utan en mardröm om struktur av kod! Framförlat hur kategorier och produkter sparas och är en stor smet mitt i GUI klassen. Halvt sovande skrev du ihop följande:

```
import java.util.Collection;

public class varoRegister {

    public void addVara(String kattergori, Vara vara){
        //TODO
    }

    public Collection<Vara> getVaror(String kattergori){
        //TODO
    }
}
```

```

public String findLKategori(Vara vara){
    //TODO - möjligen eller senare
}

public void remove(Vara vara){
    //TODO möjligen eller senare
}

public void remove(String kategori, Vara vara){
    //TODO möjligen eller senare
}

public Collection<Vara> getAllVaror(){
    //TODO - möjligen eller senare
}

public Collection<Vara> getAllKategorier(){
    //TODO - möjligen eller senare
}
}

```

Se till att dra nytta av det eller skriv ihop en egen fin wrapper för sparandet av vara i kategorier. Samt byt ut datasamlingen som du skapade innan.

Uppgift 17:

När detta nu är klart måste det tittas på GUI så att de även ska stödja de nya kategorierna. Där alla varor ska plockas bort och ersättas med kategorierna från den nya datastrukturen. Samtidigt ska den "gamla" datasamlingen plockas bort så att inga varor/kategorier sparas redundant.