

TENTAMEN OOP 2017-01-14

ANVISNINGAR

Fråga 1 och 2 besvaras på det särskilt utdelade svarsbladet, övriga frågor på de vanliga svarsbladen. Du får gärna skriva på bägge sidorna, men påbörja varje uppgift på ett nytt blad. Vid inlämning ska svaren ligga sorterade efter frågornas nummer, inte efter den ordning du besvarat dem.

BETYGSÄTTNING

Tentan består av fem frågor, och varje fråga ger max 10 poäng. Maxpoängen på tentan är således 50. Betyget på tentan sätts enligt följande kriterier:

Fx – Minst 20 poäng på tentan. Betyget kan också ges om man får minst 25 poäng, men mindre än 8 poäng sammanlagt på fråga 1 och 2.

E – Minst 8 poäng sammanlagt på fråga 1 och 2, samt minst 25 poäng totalt.

D – Minst 8 poäng sammanlagt på fråga 1 och 2, samt minst 30 poäng totalt.

C – Minst 8 poäng sammanlagt på fråga 1 och 2, minst 35 poäng totalt, samt ingen uppgift med 0 poäng.

B – Minst 40 poäng totalt, samt ingen uppgift med 0 poäng.

A – Minst 45 poäng totalt.

HJÄLPMEDEL

Inga hjälpmedel förutom den syntaxsammanfattning som delas ut på tentan.

Lycka till!

Lösningförslag läggs upp i Moodle senast tre arbetsdagar efter tentatillfället.

DEL A: FLERVALSFRÅGOR OCH KODFÖRSTÅELSE

Dessa frågor besvaras på det utdelade svarsbladet, inte på något annat sätt. På detta svarsblad får du bara skriva i de därtill avsedda fälten, ingen annanstans eftersom det delvis rättas maskinellt. Anteckningar och liknande utanför avsedda fält kan göra att det inte kan rättas, eller att resultatet blir fel. Sådana svar kommer inte att rättas om. Det är alltså ingen idé att skicka med anteckningar eller textuella förklaringar till dessa frågor eftersom vi bara kommer att titta på de avsedda fälten när vi rättar.

Flervalsfrågorna besvaras genom att fylla i hela rutan för alternativet med en mörk färg, så här: , **inte** så här: . Om du bara sätter ett kryss är risken att rättningsprogrammet missar att du markerat alternativet. Om du markerar ett felaktigt alternativ och vill ta bort det så fyller du i hela den kringliggande rutan. Skulle du göra fel två gånger på samma alternativ får du be tentavakten om ett nytt exemplar av svarsbladet och flytta över alla dina svar till det.

1.F

<input checked="" type="checkbox"/> A	<input type="checkbox"/> B	<input checked="" type="checkbox"/> C	<input type="checkbox"/> D
---------------------------------------	----------------------------	---------------------------------------	----------------------------

Exempel: A kommer att tolkas som rätt, B och D som fel. Hur C tolkas går inte att säga.

Om frågan frågar efter "vilken" eller "vilket" alternativ som är rätt så finns det exakt ett alternativ som är rätt. Skulle det av misstag från vår sida vara så att två alternativ är rätt så räcker det med att markera ett av dem, men det är mer troligt att du missat något som avgör vilket alternativ som egentligen är rätt.

Om frågan istället frågar efter "vilka" alternativ som är rätt så kan noll, ett, eller flera av alternativen vara rätt, och samtliga ska vara markerade för att poäng ska ges. Man kan alltså inte få delpoäng på en sådan fråga.

FRÅGA 1 FLERVALSFRÅGOR

Denna fråga består av tio delfrågor av flervalstyp liknande de testfrågor som finns i Moodle. Varje delfråga ger noll eller en poäng.

FRÅGA 1A

Vilket värde måste variabeln `i` ha för att koden till höger ska skriva ut en rad med tre X?

A:	1	B:	2
C:	3	D:	Inget av dessa

```
switch(i) {  
  case 1:  
    System.out.print("XXX");  
  case 2:  
    System.out.print("XX");  
  case 3:  
    System.out.print("X");  
}
```

FRÅGA 1B

Vilka av följande datatyper kan automatiskt, dvs utan en explicit typomvandling, konverteras till en `long`?

A: byte	B: double	C: int	D: String
---------	-----------	--------	-----------

FRÅGA 1C

Antag att du har en `ArrayList a`, som innehåller strängarna "Ett", "Två" och "Tre" i den ordningen. Vad innehåller listan efter att nedanstående kodrad körts?

```
a.add(1, a.remove(2));
```

A:	"Ett", "Två", "Tre"	B:	"Ett", "Tre", "Två"
C:	"Två", "Ett", "Tre"	D:	"Tre", "Ett", "Två"

FRÅGA 1D

Vilka av följande begrepp förekommer i koden till höger?

A:	Konstant
B:	Konstruktör
C:	Parameter
D:	Överlagring

```
private double d;  
  
public void inc() {  
    inc(0.1);  
}  
  
public void inc(double d) {  
    this.d+=d;  
}
```

FRÅGA 1E

Vad returnerar metoden till höger om den anropas med argumentet 3?

A:	3
B:	6
C:	123
D:	321

```
int rec(int i) {  
    if (i == 0)  
        return 0;  
    else  
        return rec(i - 1) * 10 + i;  
}
```

FRÅGA 1F

Vad av följande är sant om strängar i Java?

A:	Såväl ett strängobjekt som en referens till ett strängobjekt kan förändras
B:	Ett strängobjekt kan förändras, men det kan inte en referens till ett strängobjekt
C:	En referens till ett strängobjekt kan förändras, men det kan inte ett strängobjekt
D:	Vare sig ett strängobjekt eller en referens till ett strängobjekt kan förändras

FRÅGA 1G

Vilka av dessa påståenden om konstruktorer är sanna?

A:	Måste finnas flera i en klass
B:	Måste ha en returtyp
C:	Måste ha parametrar
D:	Måste heta samma sak som klassen

FRÅGA 1H

Vilka av följande namn från Javas standardbibliotek är antagligen klasser om namngivningskonventionerna följts?

A: ClassValue	B: Document	C: SecurityManager	D: XmlSchema
---------------	-------------	--------------------	--------------

FRÅGA 1I

Vilken del av nedanstående deklaration är det som gör PI till en konstant?

```
public static final double PI = 3.14;
```

A: public	B: static	C: final	D: PI = 3.14
-----------	-----------	----------	--------------

FRÅGA 1J

Koden till höger ger ibland ett NullPointerException. Hur kan villkoret skrivas om så att detta problem inte uppstår?

```
if(str.isEmpty()){  
    str = DEFAULT_STRING;  
}
```

A:	<code>str == null && str.isEmpty()</code>
B:	<code>str == null str.isEmpty()</code>
C:	<code>str.isEmpty() && str == null</code>
D:	<code>str.isEmpty() str == null</code>

FRÅGA 2 KODFÖRSTÅELSE

Denna fråga består av två delfrågor där du ska komma fram till vad som skrivs ut när koden exekveras. Du skall i dina svar vara noga med vad som skrivs på vilken rad, alltså beakta skillnaden mellan print och println.

FRÅGA 2A

```
String text = "Tenta";

for (int n = 0; n < text.length(); n++) {
    char ch = text.charAt(n);
    for (int m = 0; m < n; m++) {
        System.out.print(ch);
    }
    System.out.println();
}
```

FRÅGA 2B

```
TwoB t;

t=new TwoB();

System.out.println(t);

t=new TwoB(1);

System.out.println(t);

t=new TwoB();

System.out.println(t);
```

```
public class TwoB {

    private static int alpha;
    private int beta;

    public TwoB() {
        this(alpha++);
    }

    public TwoB(int x) {
        beta = x;
    }

    public int asInt() {
        return alpha + beta;
    }

    public String asString() {
        return "" + alpha + beta;
    }

    public String toString() {
        return asString() + asInt();
    }
}
```

DEL B: KODFRÅGOR

Denna del består av tre frågor där du själv ska skriva kod. Om inget annat sägs i själva frågan så spelar det ingen roll för betyget hur pass effektiv en lösning är så länge den fungerar och uppfyller de krav som ställs. Däremot ska grundläggande krav på god programmeringsstil följas – inom rimliga gränser. Förkortade namn och sent tillkomna inskjutna rader är alltså tillåtna.

FRÅGA 3

Skriv en metod som tar en array av tecken (char) som parameter, och som returnerar om arrayens innehåll utgör ett svenskt bilnummer av standardmodell, det vill säga sex tecken med tre bokstäver först följt av tre siffror. Bokstäverna kan inte vara någon av I, Q, V, Å, Ä eller Ö.

FRÅGA 4

Skriv ett program som frågar användaren om hur snabbt ett ljus av ett visst märke brinner och längden på ett ljus och som skriver ut hur lång tid ljuset brinner i timmar och minuter. Brinnhastigheten ska anges som ett decimaltal och längden som ett heltal. Brinntiden i timmar och minuter ska skrivas ut som två heltal. Du får själv välja om resultatet ska avrundas eller klippas, det spelar ingen roll för frågan.

Hur många cm brinner ljuset på en minut: 0,05

Hur långt är ljuset i hela cm: 20

Ljuset brinner i ca 6 timmar och 40 minuter

Bara fem av poängen på frågan ges för funktionaliteten, större delen av frågan var trots allt en av de första övningsuppgifterna på kursen. De resterande fem av poängen ges för uppdelning och användning av metoder. För full poäng måste metoduppdelningen vara genomtänkt och både parametrar och returvärden användas. Uppgiften är inte lämplig att lösa på ett objektorienterat sätt, så det behövs bara en klass och metoderna kan vara statiska.

FRÅGA 5

Sommaren 2016 drabbades världen av Pokémon GO-feber och alla verkade vara ute och leta efter digitala monster och samtidigt utgöra en fara för både sig själva och andra om man fick tro media. Den värsta hypen har nu lagt sig, och det är därför dags att slår mynt av detta intresse med vårt eget spel där man kan samla digitala nallebjörnar. Varför nallebjörnar? Tja, alla gillar väl nallebjörnar, och det är ju något helt annat än monster, så det är unikt. Eller något sådant, huvudsaken är att vi kan få folk att ~~beta~~ ha roligt.

Din uppgift består av två delar där den första är att skriva början på en klass som representerar en nallebjörn. Denna klass har två attribut vi är intresserade av just nu: ett namn och ett antal poäng som spelaren får när denne gör något i spelet. Bägge ska sättas när man skapar en nallebjörn. Namnet ska både gå att läsa av och ändra, medan poängen bara ska gå att läsa av. Varje gång man läser av poängen ska den dock halveras, så om en nallebjörn var värd 256 poäng första gången antalet poäng lästes av, så är den bara värd 128 nästa gång, och 64 den tredje.

Vi ska också kunna lägga till en nallebjörn i en ArrayList om det inte redan finns en nallebjörn i listan som heter samma sak. Denna funktion ligger i metoden **add** i klassen **BearCollection** vars relevanta delar syns i klassdiagrammet till höger. Den andra delen av uppgiften är att skriva metoden **add**.

BearCollection
- bears: ArrayList<Bear>
+ add(b: Bear): void

TENTAMEN OOP 2017-04-01

ANVISNINGAR

Fråga 1 och 2 besvaras på det särskilt utdelade svarsbladet, övriga frågor på de vanliga svarsbladen. Du får gärna skriva på bägge sidorna, men påbörja varje uppgift på ett nytt blad. Vid inlämning ska svaren ligga sorterade efter frågornas nummer, inte efter den ordning du besvarat dem.

BETYGSÄTTNING

Tentan består av fem frågor, och varje fråga ger max 10 poäng. Maxpoängen på tentan är således 50. Betyget på tentan sätts enligt följande kriterier:

Fx – Minst 20 poäng på tentan. Betyget kan också ges om man får minst 25 poäng, men mindre än 8 poäng sammanlagt på fråga 1 och 2.

E – Minst 8 poäng sammanlagt på fråga 1 och 2, samt minst 25 poäng totalt.

D – Minst 8 poäng sammanlagt på fråga 1 och 2, samt minst 30 poäng totalt.

C – Minst 8 poäng sammanlagt på fråga 1 och 2, minst 35 poäng totalt, samt ingen uppgift med 0 poäng.

B – Minst 40 poäng totalt, samt ingen uppgift med 0 poäng.

A – Minst 45 poäng totalt.

HJÄLPMEDEL

Inga hjälpmedel förutom den syntaxsammanfattning som delas ut på tentan.

Lycka till!

Lösningförslag läggs upp i Moodle senast tre arbetsdagar efter tentatillfället.

DEL A: FLERVALSFRÅGOR OCH KODFÖRSTÅELSE

Dessa frågor besvaras på det utdelade svarsbladet, inte på något annat sätt. På detta svarsblad får du bara skriva i de därtill avsedda fälten, ingen annanstans eftersom det delvis rättas maskinellt. Anteckningar och liknande utanför avsedda fält kan göra att det inte kan rättas, eller att resultatet blir fel. Sådana svar kommer inte att rättas om. Det är alltså ingen idé att skicka med anteckningar eller textuella förklaringar till dessa frågor eftersom vi bara kommer att titta på de avsedda fälten när vi rättar.

Flervalsfrågorna besvaras genom att fylla i hela rutan för alternativet med en mörk färg, så här: , **inte** så här: . Om du bara sätter ett kryss är risken att rättningsprogrammet missar att du markerat alternativet. Om du markerar ett felaktigt alternativ och vill ta bort det så fyller du i hela den kringliggande rutan. Skulle du göra fel två gånger på samma alternativ får du be tentavakten om ett nytt exemplar av svarsbladet och flytta över alla dina svar till det.

1.F A B C D

Exempel: A kommer att tolkas som rätt, B och D som fel. Hur C tolkas går inte att säga.

Om frågan frågar efter "vilken" eller "vilket" alternativ som är rätt så finns det exakt ett alternativ som är rätt. Skulle det av misstag från vår sida vara så att två alternativ är rätt så räcker det med att markera ett av dem, men det är mer troligt att du missat något som avgör vilket alternativ som egentligen är rätt.

Om frågan istället frågar efter "vilka" alternativ som är rätt så kan noll, ett, eller flera av alternativen vara rätt, och samtliga ska vara markerade för att poäng ska ges. Man kan alltså inte få delpoäng på en sådan fråga.

FRÅGA 1 FLERVALSFRÅGOR

Denna fråga består av tio delfrågor av flervalstyp liknande de testfrågor som finns i ILearn. Varje delfråga ger noll eller en poäng.

FRÅGA 1A

Vad blir resultatet när följande kodrader exekveras?

```
String[] s = new String[3];  
System.out.println(s[3]);
```

A:	Den tomma strängen skrivs ut	B:	<code>null</code> skrivs ut
C:	Texten <code>s[3]</code> skrivs ut	D:	Inget, koden kraschar vid körning

FRÅGA 1B

Antag att vi har en `ArrayList a` innehållandes strängarna "ETT" och "TVÅ" i den ordningen. Vad kommer `a` att innehålla efter detta anrop: `a.get(1).toLowerCase();`

A: "ett", "två"	B: "ett", "TVÅ"	C: "ETT", "två"	D: "ETT", "TVÅ"
-----------------	-----------------	-----------------	-----------------

FRÅGA 1C

Vilka av följande anrop på `nextInt`-metoden i klassen `Random` ger ett värde i intervallet -2 till +2?

A: <code>rnd.nextInt(2) - 2</code>	B: <code>rnd.nextInt(3) - 2</code>
C: <code>rnd.nextInt(4) - 2</code>	D: <code>rnd.nextInt(5) - 2</code>

FRÅGA 1D

Att jämföra om två strängar är lika i Java kan göras både med `==` och med `equals`-metoden. Vilken är skillnaden mellan dessa sätt?

A:	Det är ingen skillnad, de gör samma sak
B:	<code>==</code> jämför om det är samma objekt, <code>equals</code> om innehållet är likadant
C:	<code>==</code> jämför om innehållet är likadant, <code>equals</code> om det är samma objekt
D:	<code>==</code> jämför om innehållet är exakt likadant, <code>equals</code> ignorerar skillnaden på stora och små bokstäver

FRÅGA 1E

Givet klassen till höger och dessa två variabler:

`KlasseE x, y;`

vilket värde kommer variabeln `x.z` ha efter följande tilldelningar:

`x = new KlasseE('A');`

`x.z='B';`

`y = new KlasseE('C');`

`y.z='D';`

```
public class KlasseE {  
  
    char z;  
  
    public KlasseE(char c) {  
        z=c;  
    }  
  
}
```

(Svaret du ska ange är alltså värdet på variabeln `x.z` efter den sista tilldelningen.)

FRÅGA 1F

När du skriver en egen klass i Java så ... (inget alternativ korrekt)

A:	måste den ha en main-metod
B:	måste den ha exakt en konstruktör
C:	måste den ha minst en metod
D:	måste den ha minst en variabel

FRÅGA 1G

Om `a` och `b` är två boolska variabler, vilka av nedanstående uttryck är ekvivalenta med `!a || b` ?

A:	<code>a && !b</code>	B:	<code>!(a && !b)</code>
C:	<code>!(a b)</code>	D:	<code>!a (a b)</code>

FRÅGA 1H

Här nedanför ser vi fyra försök att läsa in ett tal större eller lika med noll. Vilket eller vilka av dessa fungerar som avsett?

A:	<pre>do { System.out.print("Tal >= 0: "); int i = readInt(); } while (i < 0);</pre>
B:	<pre>int i= readInt(); do { System.out.print("Tal >= 0: "); } while (i < 0);</pre>
C:	<pre>int i; do { System.out.print("Tal >= 0: "); i = readInt(); } while (i < 0);</pre>
D:	<pre>int i; do { System.out.print("Tal >= 0: "); int i = readInt(); } while (i < 0);</pre>

FRÅGA 1I

Vilka av följande programsatser sätter a till 7?

A:	<pre>int a = 7%13;</pre>	B:	<pre>int b = 8; int a = b--;</pre>
C:	<pre>int a = (int)6.5;</pre>	D:	<pre>int a = (int)7.5;</pre>

FRÅGA 1J

Vilka av följande påståenden är sanna om statiska variabler?

A:	Deklareras i en metod	B:	Heter samma sak som klassen
C:	Varje objekt får sin egen uppsättning av dessa	D:	Delas mellan alla objekt

FRÅGA 2 KODFÖRSTÅELSE

Denna fråga består av två delfrågor där du ska komma fram till vad som skrivs ut när koden exekveras. Du skall i dina svar vara noga med vad som skrivs på vilken rad, alltså beakta skillnaden mellan `print` och `println`.

FRÅGA 2A

```
class Zyd1 {  
  
    private String n;  
    private int a;  
    private static int tv;  
  
    public Zyd1(String n, int v) {  
        this(n, v, 1);  
    }  
  
    public Zyd1(String n, int v, int a) {  
        this.n = n;  
        this.a = a;  
        tv += a * v;  
    }  
  
    public String toString() {  
        return n + a + tv;  
    }  
  
}
```

PROGRAMMET

```
Zyd1 z;  
  
z = new Zyd1("ABC", 200, 5);  
System.out.println(z);  
z = new Zyd1("DEF", 45);  
z = new Zyd1("ABC", 200, 5);  
System.out.println(z);
```

KORREKT SVAR

ABC51000
ABC52045

FRÅGA 2B

```
System.out.print("knockknock");

for (int n = 0; n < 8; n++) {
    if (n % 2 == 0 ) {
        System.out.println();
    }
    switch (n) {
        case 1:
            System.out.print("sthere");
            break;
        case 3:
            break;
        case 2:
        case 4:
        case 6:
            System.out.print("to");
            break;
        default:
            System.out.print("who");
    }
}

System.out.println("m");
```

KORREKT SVAR

knockknock
whosthere
to
towho
towhom

(MED SKILJETECKNEN UTSATTA)

Knock-knock.
Who's there?
To.
To who?
To whom!

DEL B: KODFRÅGOR

Denna del består av tre frågor där du själv ska skriva kod. Om inget annat sägs i själva frågan så spelar det ingen roll för betyget hur pass effektiv en lösning är så länge den fungerar och uppfyller de krav som ställs. Däremot ska grundläggande krav på god programmeringsstil följas – inom rimliga gränser. Förkortade namn och sent tillkomna inskjutna rader är alltså tillåtna.

FRÅGA 3

Den första april 2010 plockade Google bort alla vokaler från Gmail och skyllde på ett serverfel som börjat med att bokstaven A inte kunnat visas för vissa användare. Problemet hade sedan eskalerat, och alla vokalerna hade drabbats. Det officiella meddelandet nämnde också att man undersökte rapporter om att även Y var drabbat.

Eftersom det är den första april idag vill vi knyta an till detta aprilskämt och uppgiften är att skriva en metod som tar in en sträng som parameter, och som returnerar strängen utan vokalerna. Om vi till exempel skickar in "APRIL" ska vi alltså få tillbaka "PRL".

För den som känner sig osäker så är vokalerna i det svenska alfabetet A, E, I, O, U, Y, Å, Ä och Ö. I det engelska alfabetet så är A, E, I, O och U vokaler, och Y verkar inte vara någon slags mellanläge. Det spelar ingen roll för uppgiften vilken av dessa uppsättningar du plockar bort, så ta bort A, E, I, O och U. Då behöver du inte skriva lika mycket kod. Av samma anledning kan du anta att strängen består av enbart stora bokstäver.

TRE LÖSNINGSVARIANTER

```
// https://en.wikipedia.org/wiki/Disemvoweling
```

```
public static String disemvowel1(String s) {  
    String result = "";  
    for (int n = 0; n < s.length(); n++) {  
        if (!isntVowel(s.charAt(n))) {  
            result += s.charAt(n);  
        }  
    }  
    return result;  
}
```

```
private static boolean isntVowel(char c) {  
    switch (c) {  
        case 'A':  
        case 'E':  
        case 'I':  
        case 'O':  
        case 'U':  
            return false;  
        default:  
            return true;  
    }  
}
```

```

    }
}

public static String disemvowel2(String s) {
    return s.replaceAll("A", "").replaceAll("E", "").replaceAll("I",
    "").replaceAll("O", "").replaceAll("U", "");
}

private static final String[] VOWELS = { "A", "E", "I", "O", "U" };

public static String disemvowel3(String s) {
    for (String vowel : VOWELS) {
        s = s.replaceAll(vowel, "");
    }
    return s;
}

```

FRÅGA 4

Skriv en klass som representerar en rektangel. Klassen ska ha fyra heltalsvariabler som representerar koordinaterna för det övre vänstra hörnet samt bredden och höjden av rektangeln. Samtliga dessa variabler måste sättas när man skapar en instans av klassen och kan sedan inte ändras. Däremot ska de gå att läsa av.

Klassen ska också ha en metod som tar en rektangel som parameter och som returnerar om denna rektangel har samma area som den rektangel som metoden anropas på eller inte.

Det ska dessutom gå att skriva ut objekt av klassen på Javas standardsätt. Det ingår i frågan att veta vad detta betyder, men resultatet ska vara något i den här stilen: (x=10, y=20, b=30, h=40).

LÖSNINGSFÖRSLAG

Halva lösningsförslaget till denna uppgift fanns i syntaxsammanfattningen. Punkt-klassen är väldigt lik.

```

public class Rectangle {

    private int x, y, width, height;

    public Rectangle(int x, int y, int width, int height) {
        super();
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
    }
}

```

```
public int getX() {
    return x;
}

public int getY() {
    return y;
}

public int getWidth() {
    return width;
}

public int getHeight() {
    return height;
}

public int getArea() {
    return width * height;
}

public boolean hasSameArea(Rectangle r) {
    return getArea() == r.getArea();
}

public String toString() {
    return "(x=" + x + ", y=" + y + ", w=" + width + ", h=" + height +
    ")";
}
}
```

FRÅGA 5

Klassdiagrammet till höger visar en kraftigt förenklad modell över några av klasserna man skulle behöva för att implementera programvaran i en bankomat. Du kan anta att alla konstruktörer och metoder i diagrammet redan finns och fungerar förutom **skrivUtKontoutdrag** i klassen Bankomat.

Samtliga instansvariabler i diagrammet är privata och samtliga konstruktörer och metoder är publika.

Din uppgift är att skriva metoden **skrivUtKontoutdrag**. Parametern metoden tar är numret på kontot för vilket kontoutdraget ska skrivas ut. Det är inte säkert att det finns något sådant konto. Om det inte gör det ska ett felmeddelande skrivas ut istället.

Om kontot finns så ska kontonummer, saldo och de fem senaste uttagen skrivas ut på **System.out** i enlighet med exemplet nedan:

Konto nummer 38726119

Saldo 7300

2017-04-01 200 kronor

2017-03-23 1200 kronor

2017-03-18 100 kronor

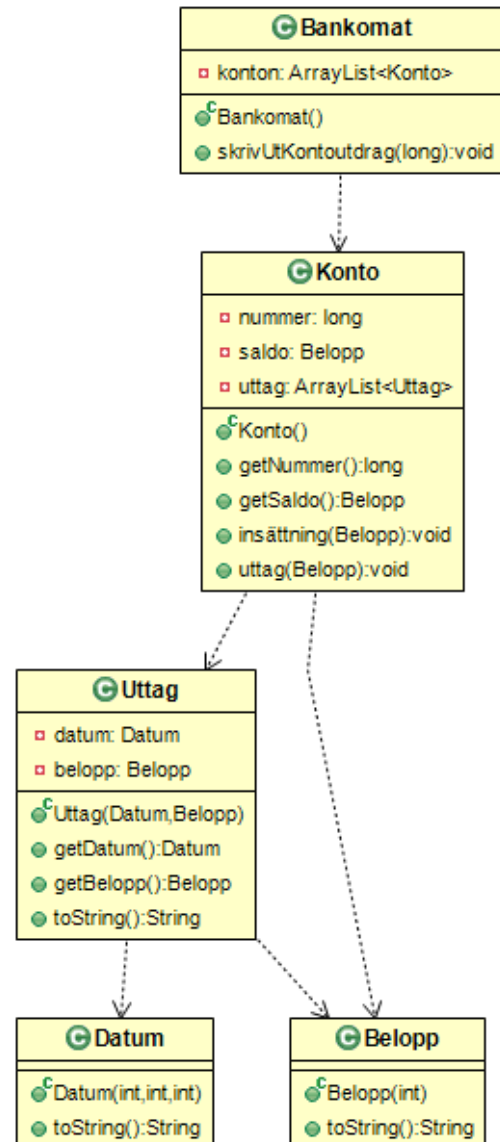
2017-03-12 2300 kronor

2017-03-02 800 kronor

Uttagen ligger sorterade efter datum i listan **uttag**, med det äldsta uttaget först och det senaste sist. Tänk dock på att det inte behöver finnas fem uttag i listan. Kontot kanske är helt nyskapat första gången kunden begär ett kontoutdrag.

toString-metoderna i klasserna **Datum** och **Belopp** formaterar texten korrekt enligt exemplet ovan.

Du får inte ändra på något av det som redan finns i klassdiagrammet, men du får lägga till fler metoder i klasserna om det skulle behövas. Dessa metoder måste du då skriva koden för själv. Om du lägger till metoder på detta sätt så måste de antingen vara privata eller ha returtypen void. Du får alltså inte lägga till några publika metoder som returnerar något.



LÖSNINGSFÖSLAG

```
// I klassen bankomat.
public void skrivUtKontoutdrag(long nummer) {
    Konto k = sökKonto(nummer);

    if (k != null) {
        k.skrivUtKontoutdrag();
    } else {
        System.out.println("Det finns inget konto med det numret");
    }
}

// I klassen bankomat.
private Konto sökKonto(long nummer) {
    for (Konto k : konton) {
        if (k.getNummer() == nummer) {
            return k;
        }
    }

    return null;
}

// I klassen konto. Begränsningarna i uppgiften gör att vi inte kan
// returnera hela eller delar av listan, så vi måste göra utskriften i den
// här klassen.
public void skrivUtKontoutdrag() {
    System.out.println("Konto nummer " + nummer);
    System.out.println("Saldo " + saldo);

    for (int i = uttag.size() - 1; i >= 0 && i >= uttag.size() - 5; i--) {
        Uttag u = uttag.get(i);
        System.out.println(u.getDatum() + " " + u.getBelopp() + " kronor");
    }
}
```