

Tentamen

Instruktioner

Besvara följande uppgifter så väl som möjligt. Om något skulle vara oklart, gör en notering kring detta i anslutning till lösningen. Om du har glömt eller är osäker på namnet på någon metod/klass, ange då ett rimligt namn, och gör en notering om att du gjort det, och vad metoden/klassen har för funktion.

Bedömning

Varje uppgift kan ge max 10 poäng

G-gräns: 40 poäng

VG-gräns: 60 poäng

Uppgift 1

Skriv kod för metoderna

```
public void insert(E data)
public E removeFirst() throws NoSuchElementException
```

som opererar på en enkellänkad lista. `insert` sätter in elementet `data` först i listan. `removeFirst` tar bort och returnerar det första elementet i listan, eller kastar ett undantag om listan inte innehåller något element. Utgå från följande kod

```
public class MyLinkedList<E> {
    private class Node {
        E data;
        Node next;

        public Node(E data, Node next) {
            this.data = data;
            this.next = next;
        }
    }
    private Node first = null;
}
```

Implementerar den här länkade lista någon av de abstrakta datatyperna stack eller kö, och i så fall vilken?

Uppgift 2

Förklara begreppet arv och ge ett kodexempel. I ditt kodexempel ska det finnas konstruktörer för en superklass och en subclass, där subclassens konstruktor anropar superklassens.

Uppgift 3

Beskriv hur linjär respektive binär sökning utförs. Diskutera när algoritmerna kan användas och deras tidskomplexitet. Relatera även användningen av sökalgoritmerna till datastrukturerna array och länkad lista.

Uppgift 4

Beskriv ett binärt sökträd. Hur är det strukturerat, och hur går insättning respektive sökning till? Vilken tidskomplexitet har dessa operationer om trädet är balanserat respektive obalanserat? Visa även hur ett träd skulle se ut om man sätter in nycklarna 6, 2, 5, 7, 4, 10 i den ordningen, samt hur en sökning efter 4 respektive 1 går till.

Uppgift 5

- a) Vad innebär synlighet? Varför vill man kunna begränsa synligheten på variabler och metoder?
- b) Var någonstans kan man komma åt de fyra variablerna i klassen nedan? Antag att Class1 har en subclass SubClass1, och att det finns fler klasser i paketet myPackage.

```
package myPackage;
public class Class1 {
    private int v1;
    protected int v2;
    public int v3;
    int v4;
}
```

Uppgift 6

- a) Skriv en (icke-abstrakt) klass som ärver från den abstrakta klassen Class2 nedan. Metoden `inOrder(int[])` ska kontrollera om talen i en array ligger i stigande ordning. Lika tal kan förekomma, och räknas som stigande.

```
public abstract class Class2 {
    public abstract boolean inOrder(int[] values);

    public void helloWorld {
        System.out.println("Hello world!");
    }
}
```

- b) Hade det gått att använda ett gränssnitt ("interface") istället för en abstrakt klass? Vad hade i så fall behövt ändras i koden (både den givna och din egen)? Vad skiljer ett gränssnitt från en abstrakt klass?

Uppgift 7

- a) Vilka strängar matchar följande reguljära uttryck

1. `histori([ae]n?|er(na)?)s?`
2. `[Hh]oppa([rt]|de)?`

- b) Skriv ett reguljärt uttryck som matchar följande böjningsformer av "fin": "fin", "fint", "fina", "finare", "finast" och "finaste". Ditt uttryck ska vara så kort som möjligt.

- c) Rita en finit automat för uttrycket i uppgift b. Använd så få tillstånd som möjligt och ha ett tecken per båge.

- d) Skriv en metod som hittar och skriver ut alla ord som börjar och slutar på samma bokstav. Din metod ska ta emot en sträng som den matchar i, och skriva ut alla sådana ord till `System.out`. Bortse från versaler och utgå från att texten endast innehåller gemener. Om man skickar in texten "anna anses vara girig" ska "anna" och "girig" skrivas ut. För full poäng ska matchningen göras med hjälp av ett reguljärt uttryck.

Uppgift 8

Beskriv principiellt hur man gör när man lägger till ett element sist i en lista, implementerad med hjälp av en array respektive som en länkad struktur. Anta att den länkade strukturen är en enkellänkad lista med en referens till första objektet. Vilken tidskomplexitet får insättning för de två strukturerna i värsta fallet? Skulle man kunna förbättra tidskomplexiteten för den länkade listan på något sätt, och i så fall hur?