

**Tentamen IK202G/IK2013, Informatik B, Objektorienterad programmering med C#, 7,5 högskolepoäng**

Delkurs: Provkod: 0110, Skriftlig tentamen, 6 högskolepoäng

Datum: [2017-08-19]

Antal timmar: [4]

Ansvarig lärare: Mevludin Memedi, Ann-Sofie Hellberg

Antal frågor: [10]

Maxpoäng: [20]

Godkänd: För betyget Godkänt (G) krävs minst 50 % av den totala poängen.

Väl godkänd: För betyget Väl godkänt (VG) krävs minst 75 % av den totala poängen.

Anvisningar:

Hjälpmedel: Ordbok till/från svenska och annat språk. Ordbok till/från engelska och annat språk.

Skriv din tentamenskod på varje papper som Du lämnar in.

Skriv endast på ena sidan av pappret.

Klargör dina [beräkningar/argument/antaganden] tydligt!

Är uppgiften oklar, gör rimliga antaganden och redovisa dessa.

Skriv LÄSLIGT!

Tentamensresultat:

Tentamensresultatet meddelas via Studentforum.

**LYCKA TILL!**

**Fråga 1 (2p)**

Vad uppnås genom följande kod och vad är skillnaden mellan varA och varB?

private string varA;

public string VarA

{

get

{

return varA;

}

set

{

varA = value;

}

}

public string varB;

**Fråga 2 (2p)**

1. Vad är användningsområdet för partiella klasser? (1p)
2. Ge ett exempel på ovanstående genom att skriva kod (1p)

**Fråga 3 (1p)**

Förklara skillnaden mellan de två medlemmarna i ClassA nedan.

class ClassA

{

protected string stringA;

internal string stringB;

}

**Fråga 4 (1p)**

Beskriv kortfattat den huvudsakliga skillnaden mellan arv och gränssnitt (1p).

**Fråga 5 (1p)**

Ange om datatypen är av värde- eller referenstyp för följande nyckelord genom att sätta kryss för rätt alternativ (skriv av tabellen på ditt svarsblad).

|  |  |  |
| --- | --- | --- |
| **Nyckelord** | **Värdetyp** | **Referenstyp** |
| int |  |  |
| interface |  |  |
| string |  |  |
| double |  |  |
| class |  |  |

**Fråga 6 (2p)**

1. Förklara syftet med överlagring av metoder (1p)
2. Skriv ett kodexempel för en överlagring av en metod (1p)

**Fråga 7 (5p)**

Skriv ett exempel med återanvändbar kod för att modellera nedanstående klasshierarki. Använd inte gränssnitt i din lösning.

Hierarkin ska modellera information om två studenttyper, campusstudenter (CampusStudent) och distansstudenter (OnlineStudent). Alla typer av studenter skall spara information om namn och personnummer men när det kommer till beteendet skiljer det sig mellan olika typer av studenter. Campusstudenter läser kurser genom att delta i föreläsningar i fysiska klassrum medan distansstudenter använder lärplattformar via Internet. Modellera detta beteende genom användning av polymorfism. Fokus skall ligga på att modellera beteendet, inte på funktionaliteten i den faktiska implementationen. Det är således tillräckligt om de olika beteendena exemplifieras genom till exempel olika utskrifter på skärmen. Observera dock att det *inte* ska vara möjligt att instansiera objekt av typen Student.

Student

CampusStudent

OnlineStudent

**Fråga 8 (2p)**

1. Skriv kod så att klassen Student i fråga 7 nu istället är ett gränssnitt (1p)
2. Vad blir konsekvensen av en sådan förändring? (1p)

**Fråga 9 (2p)**

1. Förklara nedanstående kod genom att beskriva vad som händer rad för rad och vad syftet med koden är (1p)
2. Förklara huvudskillnaden mellan att spara data så som sker i exemplet jämfört med att spara data i en textfil (1p)

Student studentObject = new Student();

FileStream outFile = new FileStream("StudentBinaryData", FileMode.Create, FileAccess.Write);

BinaryFormatter bFormatter = new BinaryFormatter();

bFormatter.Serialize(outFile, studentObject);

outFile.Close();

// Skriv kommentar... (0.5p)

FileStream inFile = new FileStream("StudentBinaryData", FileMode.Open, FileAccess.Read);

Student studentObject2 = new Student();

studentObject = (Student)bFormatter.Deserialize(inFile);

inFile.Close();

// Skriv kommentar... (0.5p)

**Fråga 10 (2p)**

1. Skriv kod för att skapa en samling där objekt av de två typerna av studenter (fråga 7) kan lagras. Lägg till åtminstone 2 objekt av varje klass, dvs. två objekt av typen CampusStudent och två objekt av typen OnlineStudent. (1p)
2. Skriv kod som itererar samlingen och kallar på metoden för motsvarande objekt (1p)