(1)

Namn | Personnummer | Blad nr | Uppgift nr

1. a) PRIVATE:
åtkomst endast inom den egna klassen.
private är den högsta skyddsnivån

b) PROTECTED:
åtkomst inom den egna klassen, i klasser i det egna paketet och eventuella subklasser i andra paket. (bra vid arv)

c) PUBLIC - åtkomst överallt i programmet.
public är den lägsta skyddsnivån.

Om man inte anger någon modifierare blir det automatiskt package, Bra!
dvs åtkomst/synlighet endast i klasser inom det aktuella paketet.

3

Mycket bra svar på samtliga frågor! 48

AR

VG

(2)

KTH VETENSKAP OCH KONST

Personnummer | Blad nr

2. Static markerar klasstillhörighet (& inte instanstillhörighet)
. När man sätter static framför en variabel blir det en klassvariabel.
klassvariabler är gemensamma för alla objekt som tillhör klassen. Det finns bara en av den och värdet är detsamma för alla objekt som tillhör klassen.
(En metod som markeras static blir en klassmetod. En klassmetod hanterar inga individuella objekt utan använder bara klassvariabler.)

(Dvs alla instanser som skapat av en klass som markerats som static kommer att dela på värdet.)

(En klassmetod anropas enligt följande:
klassnamn.metodnamn(argument))

heltalsvariabler får typen Int dvs int står för heltal.

2

3

Personnummer | Blad nr

3). x = 9 i = 9 initiering

en forloop består av (~~iteration~~, villkor och förändring)

forloopen säger:

så länge som i är större eller lika med 0

sua (i) minskas med 1 (i--)

(i-- betyder att i ska användas först innan det minskas med ett)

Utskriften kommer att bli: i * x: _resultat

för varje utskrift kommer man hoppa till nästa rad (System.out.println)

i = 9 x = 9 i = 8 x = 9 i = 7 x = 9

VARV 1: 9 * 9: _81 VARV 2: 8 * 9: _72 VARV 3: 7 * 9: _63

i = 6 x = 9 i = 5 x = 9 i = 4 x = 9

VARV 4: 6 * 9: _54 VARV 5: 5 * 9: _45 VARV 6: 4 * 9: _36

i = 3 x = 9 i = 2 x = 9 i = 1 x = 9

VARV 7: 3 * 9: _27 VARV 8: 2 * 9: _18 VARV 9: 1 * 9: _9

i = 0 x = 9

VARV 10: 0 * 9: _0

Nästa varv blir i = -1 och då gäller inte villkoret längre.

6

PÅ NÄSTA BLAD FÖLJER UTSKRIFTEN

VAR VÄNLIG VÄND TILL NÄSTA BLAD →

(4)

Namn | Personnummer | Blad nr | Uppgift nr

FORTSÄTTNING FRÅGA (3)

Utskriften kommer att bli:

9 * 9: _81
8 * 9: _72
7 * 9: _63
6 * 9: _54
5 * 9: _45
4 * 9: _36
3 * 9: _27
2 * 9: _18
1 * 9: _9
0 * 9: _0

(Dvs 9:ans tabell)

_ står för tabb = \t
alltså att det blir ett mellanrum mellan : och resultat.

5

Personnummer | Blad nr

4). public class Built {

Klassvariabel, alla objekt som skapas av klassen kommer att få värdet 0.

BUILT

static int counter = 0

Separata räknare i de båda ärvande klasserna istället

public String toString()
return
"This is a built objekt.";

DÅ JAG INTE ANGETT NÅGON MODIFIERARE (SKYDDSNIVÅ) PÅ DE OLIKA VARIABLERNA FÅR DE AUTOMATISKT SKYDDSNIVÅN PACKAGE -

dvs - åtkomst endast i klasser i det egna paketet.

EXTENDS

HOUSE

double yta;

public String toString()
return;
"This is house number:"
+ counter;

EXTENDS

BICYCLE

int antalVäxlar
boolean arEnTandem

om boolean inte tilldelas något värde blir den falsk vilket i detta fall är mest troligt då de flesta bicycles (cyklar) inte är tandemcyklar.

6

KONSTRUKTOR:

```
public Built(){
  ++counter;
}
```

För varje objekt som skapas av klassen kommer counter att öka med ett och på så sätt får varje objekt ett "eget" nummer.

Alla variabler förutom conter är instansvariabler

int = står för heltal, tal utan decimaler

double = flyttal, tal m. decimaler (15 siffrors noggrannhet)

house och bicycle ärver built.

I javakod skulle det se ut:

```
public class House extends Built {
public class Bicycle extends Built {
```

Extends = ärva en klass ut från en annan. Man kan bara ärva ut av en klass.

(6)

KTH VETENSKAP OCH KONST

Personnummer Blad nr

5. Överskuggning är när en metod i superklassen omdefinieras för att passa subklassen. subklassens metod överskuggar då metoden i superklassen.
I föregående uppgift omdefinierades superklassens metod just för att den skulle passa för de två subklasserna.
(När en metod överskuggar en annan är det alltså för att metoden ska passa för en viss klass.)
Dynamisk bindning är när rätt metod automatiskt körs vid exekvering.
Javainterpretatorn börjar längst ner i arvshierarkin och söker sedan uppåt till den hittar en metod som matchar de angivna parametrarna.

(Vid dynamisk bindning har metoderna samma namn & signatur men de måste ligga i olika klasser.)

(Även superklassens metod är en överskuggning av metoden String toString)

toString() i klassen Built är en överskuggning av den toString som finns i klassen Object

Mycket bra svar!

4

(7)

KTH VETENSKAP OCH KONST

Personnummer Blad nr

6.) selektion är när vi i våra program ställs för olika vägval.
Vi behöver sätt att styra våra vägval. Detta kan vi göra genom att använda oss av if-else & switchsater.
Switchsatser passar bäst när vi har många vägval (fler än 3)
en switchsats är uppbyggt av olika case, vilka tilldelas varsitt värde. Finns inte värdet med bland de olika case-alternativen går man vidare till default-alternativet om det finns ngt
if,else kan se ut på följande vis:
tex: if(x>5){
säga följande hända
else..
(etc.)

6

Iteration betyder upprepning och i java görs iteration med programslingor (loopar) hur många gånger en kod ska itereras bestäms av ett villkor.

VÄND TILL NÄSTA BLAD ⟶

(8)

Namn Personnummer Blad nr Uppgift nr

FORTSÄTTNING FRÅGA (6)

villkor ställs med följande konstruktioner:

while - villkoret testas först

do-while - fungerar som while men villkoret testas sist

for-loop - passar bäst när man vet hur många gånger koden ska itereras.
for (initiering; villkor; förändring)

(9)

KTH VETENSKAP OCH KONST

Namn | Personnummer | Blad nr | Uppgift nr

7) En stack fungerar som en tallrikshög i en matbespisning.
Dvs. sist in först ut = LIFO (Last in first out)
Det objekt som lagts till sist i stacken är också det man tar ut först.

sist in blir också först ut.
man kan använda sig av funktionerna
push (lägg till) and pop (ta ut).

Vi använde oss av stacken under våra laborationer när vi skulle konstruera ett palindrom.
(Dvs ett ord som blir samma sak framlänges som baklänges)
Vi behövde då ett sätt att jämföra ett ord bokstav för bokstav (framlänges & baklänges)
Därför behövde vi också använda oss av en kö.
Queue en kö fungerar som en stack fast tvärtom.
FIFO - first in first out → 5 4 3 2 1 de objekt man stoppat in först är också det objekt som man plockar ut först.
Genom att använda oss av kö & stack kunde vi jämföra en textsträng, framåt & bakåt.
Dvs om den blev samma sak bakåt som framåt dvs om det var ett palindrom eller inte.

4

(10)

Namn Personnummer Blad nr Uppgift nr

8). En array är ett fält/behållare för flera objekt av samma typ. Fälten är numrerade och indexeringen börjar alltid vid noll.
De olika minnesplatserna kan tilldelas egna värden oberoende av varandra.
En array måste alltid initieras.
tex: int[] heltal = new int[5]
eller: int[] heltal = {5, 8, 10, 9, 7}
(0, 1, 2, 3, 4)
arrayen tilldelas värden

Om man tex skriver:
int x = heltal[2] så menar man talet på position 2 som i detta fallet är 10.
Om ett tal finns inom [1] menar man alltid på position 1. (inte 1 som tal i sig).

4

En konstruktor känns igen enligt följande: (7)

1. Är alltid publik (public)
2. Har ingen returtyp
3. Heter alltid samma sak som klassen.

9.) En konstruktor är en slags initierings-metod som anropas automatiskt varje gång ett nytt objekt skapas av klassen.
En överlagrad konstruktor =
Det kan finnas flera konstruktorer med samma namn i en klass men de måste ha olika parametrar (dvs. argumentlistor)

→ antingen olika antal parametrar
→ eller olika typer på parametrarna.

I uppgift två ITKPerson och dess subklasser förstod jag konstruktorernas funktion och hur argument skickas med.
I baskassen använde jag mig av konstruktorer på följande sätt:

```
public ITKPerson(String namn) {
    this.namn = namn;
}
```

6

och:

```
public ITKPerson(String namn, String fnNummer)
    this.namn = namn;
    this.fnNummer = fnNummer;
}
```