Arv och polymorfism i Java

Objektorienterad programmering
Laboration 5

Syfte

Att ge en introduktion till arvsmekanismen i Java.

Mål

Efter övningen skall du kunna

- definiera klasser med arv i Java.
- förstå hur polymorfism fungerar.

Utvecklingsmiljö

BlueJ på valfri plattform.

Litteratur

Kursboken kap 1-10 i allmänhet, 8-10 i synnerhet. Begrepp: Refactoring, kodduplicering, arv, basklass (super class), subklass, abstrakt klass, abstrakt metod, statisk typ, dynamisk typ, dynamisk bindning, polymorfism, djup kopiering.

Dokumentet ”Kopiering av objekt i Java”.

Färdig programkod

Given programkod finns på kursens hemsida i **Laborationer->Programkod för labbarna ->arv.zip**.

Labbgrupper

Arbetet genomförs i grupper om **två** personer.

Redovisning

Senaste redovisningsdag, se kurs-PM samt Fire.

Ladda upp filerna `Shape,Square,Circle,Triangle,Group,NewHousePicture` samt `HouseTest` med dina lösningar i, samt `README.txt` med allmänna kommentarer om lösningen, i Fire. Redovisa slutversionerna av `Shape,Square,Circle,Triangle` och `Group` som de ser ut efter att du löst samtliga uppgifter.

Glöm inte att trycka på **SUBMIT***!*

Problembeskrivning

Projektet `arv` innehåller ett exempel med en bild på ett hus som kan ritas upp grafiskt i ett ritfönster. Exemplet liknar ett som finns i kapitel 1 i kursboken, men med några små skillnader. Vi skall *inte* använda koden i kapitel 1 i den här övningen. Problemet med koden i `arv` är att den är dåligt strukturerad och behöver förbättras. Det finns t.ex. många identiska kodavsnitt som upprepas i flera av klasserna.

Uppgifter

Börja med att öppna `arv` och studera klasserna. Uppgifterna går ut på att bygga om (refaktorera) koden genom att använda arvsmekanismen i Java.

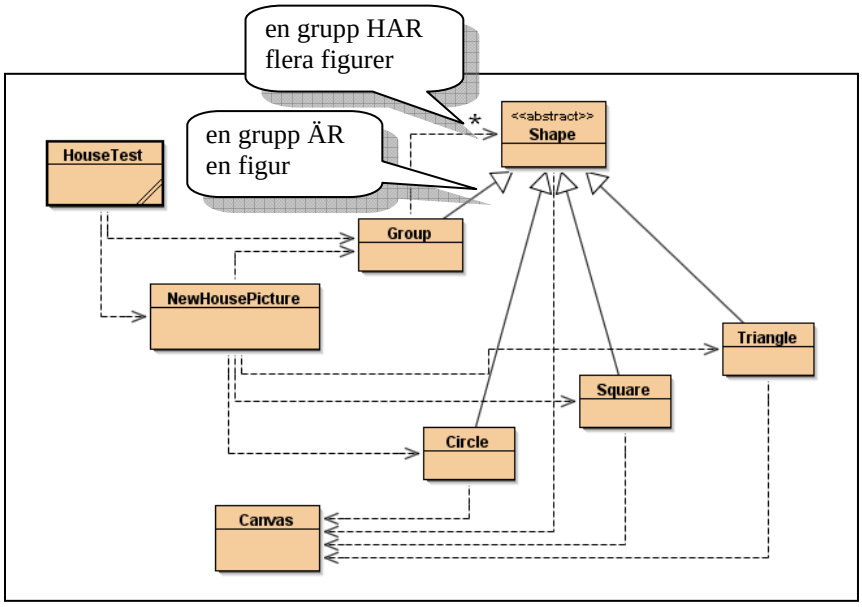
Uppgift 1 Inför en basklass till Square, Circle och Triangle

Som du ser har klasserna `Square`, `Circle` och `Triangle` mycket kod som är helt identisk. Gemensamma instansvariabler och metoder kan flyttas till en ny klass `Shape` som får vara basklass till de tre klasserna ovan. Börja gärna med att skriva ut all kod på papper så får du bättre överblick. Gå igenom koden noggrant och bestäm vad som skall flyttas till basklassen. Basklassen skall ha en lämplig konstruktor. Kom ihåg att anropa denna i subklasserna (super). Vilka parametrar måste subklassernas konstruktorer ha? Tänk efter vilka metoder som kan placeras i basklassen. Om en instansvariabel eller en metod *kan* ligga där så *skall* den göra det.
Tips: Två av metoderna måste definieras i alla subklasserna och de skall dessutom definieras som abstrakta i basklassen – som också deklareras som abstrakt. Gå inte vidare förrän du är helt säker på vilka de två metoderna är! Deklarera samtliga instansvariabler i basklassen som `protected` så att de blir synliga subklasserna.
BlueJ: Skapa basklassen med högerklick i fönstret och välj New Class, skriv in namnet och välj sedan Abstract Class. Dra arvspilar från subklasserna till den nya klassen.

Om du gjort allt rätt skall programmet fungera som tidigare efter omarbetningen.

Uppgift 2 En klass för sammansatta figurer

Hittills har varje figurobjekt motsvarat en enkel figur men man kan också tänka sig att bilda sammansatta figurer, ungefär som man grupperar figurobjekt i ritprogram och ordbehandlare. Om vi representerar sammansatta figurer med klassen `Group` så skall en sammansatt figur också betraktas som en figur, och därmed blir `Group` subklass till `Shape`. Ett objekt av klassen `Group` kan innehålla flera andra figurobjekt, även gruppobjekt, därav den streckade pilen från `Group` till `Shape` i klassdiagrammet nedan.



innan du fortsätter bör du studera papperet "Kopiering av objekt i Java"
(`kloning.pdf`)

Ett visst figurobjekt kan adderas till flera olika grupper. Man måste bestämma om ett och samma figurobjekt kan tillhöra flera grupper samtidigt eller ej. I den här övningen gäller att detta inte får ske. Alla gruppobjekt skall äga sina delfigurobjekt exklusivt. Ingen delning av objekt får förekomma. Därför börjar vi med att lägga till metoden

`public` *Figurklass* `clone()` (definiera den i samma stil som på sid. 4 i dokumentet ovan)

i klassen `Shape`. (Behövs det en `clone` i `Circle`, `Square` och `Triangle`? Tänk efter!)

Metoder i `Group`:

`clone()`	returnerar en djup kopia av gruppobjektet
`add(Shape item)`	adderar en kopia av `item` till gruppen
`draw()`	ritar upp gruppens delfigurer i ritfönstret
`erase()`	raderar gruppens delfigurer från ritfönstret
`move(dx,dy)`	flyttar hela gruppen i angiven riktning och ritar om den
`resize(scaleFactor)`	skalar om bilden proportionellt. Vid omskalning skall hela bilden "blåsas upp". Bilden får inte ramla isär. Delbilderna måste både flyttas och skalas om på lämpligt sätt. *Testa detta noga!*

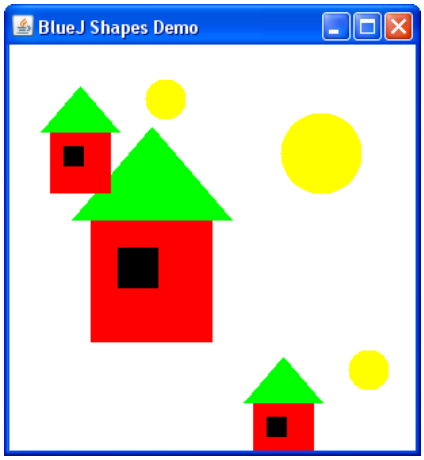
Anm. Om kompilatorn varnar för osäkra typomvandlingar vid kompileringen av `Group` kan du lägga till annotationen `@SuppressWarnings("unchecked")` före `clone`-metoden.

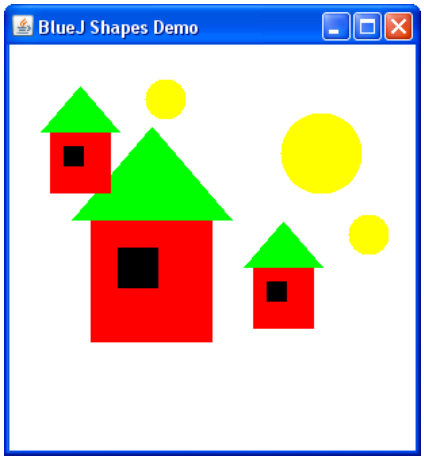
Uppgift 3 Testa figurgruppsklassen

Skriv färdigt klassmetoden `getPicture` i klassen `NewHousePicture`. Metoden skall skapa och returnera husbilden representerad som en figurgrupp. Nedanstående satser (finns i `HouseTest.java`)

```
Group house = NewHousePicture.getPicture();
house.draw();
Group house2 = house.clone();
house2.resize(0.5);
Group house3 = house2.clone();
house3.move(150,200);
```

skall resultera i bilden till vänster.





Om du istället exekverar

```
Group house = NewHousePicture.getPicture();
house.draw();
Group house2 = house.clone();
house2.resize(0.5);
Group house3 = house2.clone();
house3.move(150,200);
house3.moveTo(150,100);
```

skall du få bilden till höger. Om inte, se första tipset nedan.

Tips:

- Gruppen måste ha en ”egen” position, det räcker inte att delfigurerna har sina. Denna kan från början sättas till (0,0). En ledtråd till varför positionen behövs är att studera hur metoden `moveTo` är definierad i `Shape`. *Testa noga att moveTo fungerar korrekt för gruppobjekt!*
- Instansvariablerna för färg och synlighet (`isVisible`) kan initieras godtyckligt i anropet av basklasskonstruktorn eftersom de inte har någon direkt funktion i ett gruppobjekt.
- Vad händer om man skriver `house.add(house)`? Testa och förklara!

Lycka Till!