Oppgaver til repetisjonstime om rekursjon

Hvis du ønsker å løse oppgavene nedenfor selvstendig, kan du bruke prekoden som ligger i mappen «Prekode».

Oppgave 1

Skriv en rekursiv metode med signaturen public static void spoersmaalR(int n, Scanner scanner). Metoden skal skrive ut en beskjed til brukeren via terminalen. Denne beskjeden skal informere brukeren om hvor mange ganger den rekursive metoden har blitt kalt, og spørre brukeren om de ønsker å fortsette. Dersom brukeren ønsker å fortsette (og dermed få en ny beskjed), skal de få mulighet til dette. Hvis de ikke ønsker å fortsette, skal metoden skrive ut en beskjed som informerer brukeren om totalt antall rekursive kall, og terminere. Den første gangen vi kaller på metoden vil n være lik 0.

Oppgave 2

Klassen Vare er en enkel klasse som representerer en vare som du kan kjøpe i en dagligvarebutikk. Vare-objekter har et navn, for eksempel «eple», og en referanse til neste Vare-objekt. Klassen GlutenfriVare er en subklasse av Vare, og implementerer interface-et Allergivennlig. Klassen Dagligvarer er en enkeltlenket lenkeliste som består av objekter av klassen Vare. Den har en enkel metode leggTil(Vare vare) som legger til Vare-objekter på slutten av listen. I tillegg har den en referanse til foerste og siste Vare-objekt i listen.

Du skal nå skrive to metoder:

(1) I klassen Vare skal du skrive en rekursiv metode med signaturen public Vare finnVareR(boolean erAllergisk, String navn). Hensikten med metoden er å finne varen en bruker er på utkikk etter. Parameteren «navn» beskriver det brukeren ønsker å kjøpe, for eksempel «eple». Parameteren erAllergisk forteller oss om brukeren er allergisk eller ikke. Hvis varen er den brukeren er på utkikk etter, og brukeren ikke er allergisk, skal metoden returnere varen. Hvis brukeren derimot er allergisk, skal varen kun returneres dersom den er allergivennlig. Hvis varen ikke passer til brukerens kriterier, skal du lete videre hvis det finnes flere varer igjen i Dagligvarer.

(2) I klassen Dagligvarer skal du skrive en metode public Vare kjoepVare(boolean erAllergisk, String navn). Metoden har altså de samme parameterne som sistnevnte metode, og du skal bruke dem til å kalle på den. Metoden skal enten returnere varen som passer til brukerens kriterier, eller null hvis den ikke finner noen.

Oppgave 3

Skriv en rekursiv metode med signaturen public static ArrayList<String> plukkBlomsterR(int indeks, String[][] sti, ArrayList<String> kurv). Hensikten med metoden er å plukke (noen spesifikke) blomster langs en sti, legge dem til i kurven vår og til slutt returnere denne kurven. Metoden skal ta inn en int indeks (som skal starte på 0), en todimensjonal Array som representerer en tursti og en ArrayList som representerer en kurv. Turstien går fra et punkt A (indeks 0) til et punkt B (sti.length-1), og tilbake samme vei. I metoden skal vi gå gjennom turstien, og for hvert stopp (representert av en indre Array) skal vi plukke blomster etter følgende kriterier:

(1) Hvis vi er på vei fra A til B, og vi finner en hvitveis, skal vi plukke den og legge den til i kurven vår

(2) Hvis vi finner en gullstjerne, skal vi plukke den, legge den til i kurven vår og gå tilbake (det var denne vi aller helst ville ha!)

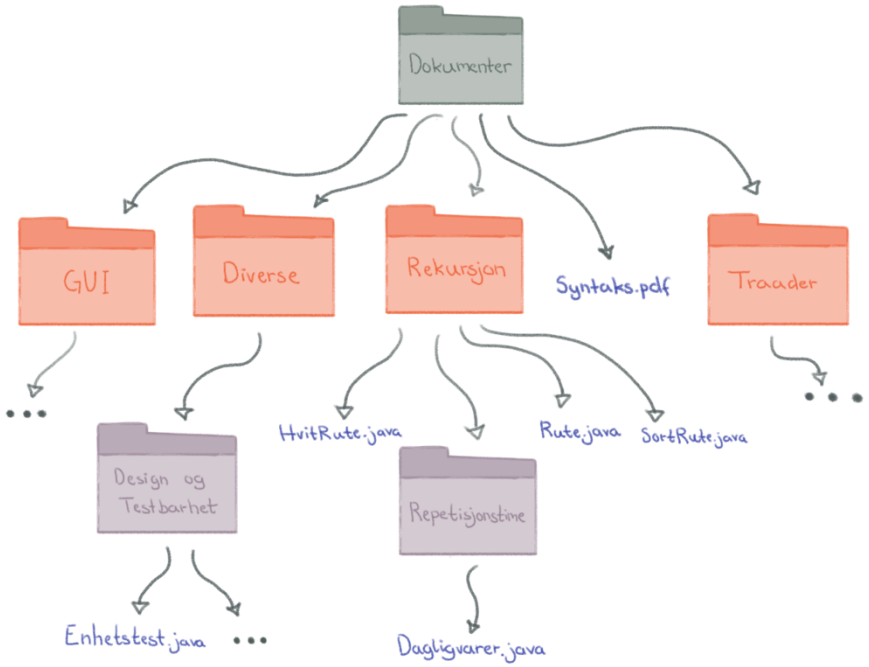
(3) Hvis vi er på vei tilbake fra B til A, og vi finner en blaaveis, skal vi plukke den og legge den til i kurven vår.

Oppgave 4

I denne oppgaven skal vi simulere et filsystem. Klassen Fil er en enkel klasse som representerer en fil. Filen har et navn og en metode som returnerer dette navnet. Klassen Mappe er en enkel klasse som representerer en mappe. Denne klassen har et navn, og to beholdere som holder rede på filer og andre mapper som denne mappen inneholder. Den har, blant annet, noen enkle metoder som gjør det mulig å legge til filer og mapper i disse beholderne. Klassen Filsystem er en graf (se bildet nederst i oppgaveteksten her) som har en referanse til en hovedmappe. Herfra skal vi kunne finne fram til alle mapper og filer i filsystemet vårt. I prekoden er det opprettet et eksempel på et slikt filsystem. Du

står selvfølgelig fri til å opprette ditt eget filsystem og utvide funksjonaliteten til klassene dersom du ønsker det. Det anbefales å tegne filsystemet før du løser oppgaven.

Skriv den rekursive metoden public String finnFilstiR(String filnavn, Mappe mappe). Metoden skal ta inn et filnavn og en mappe fra en bruker, og returnere filbanen fra mappen til filen dersom filen kan nås fra denne mappen. Hvis filen ikke finnes, skal metoden returnere null.



Bilde: Eksempel på et Filsystem med mappen «Dokumenter» som hovedmappe.