Obligatorisk oppgave 6 i INF1010, våren 2014: "Beholdere til Leger og resepter"
Versjon 1,0. Frigis 26, februar, 3 poeng. Leveres innen kl 10:00 onsdag 18. mars.

I denne oppgaven skal dere vise at dere kan bruke grensesnitt, abstrakte klasser, generiske klasser, subklasser, lenkelister og tabeller.
Dere skal programmere en del beholdere som skal brukes i oblig 7. Mange av disse beholderene skal kunne brukes til andre formål, men for å ikke gjøre oblig 6 for abstrakt er det kanskje lurt å ha bruken i oblig 7 klart for øyet hele tiden.
På gruppene vil dere få hjelp til å lage iteratorer. Når dere implementerer iteratorer behøver dere ikke implementere remove().

De beholderene dere skal programmere i oblig 6 skal i oblig 7 ta vare på alle legemidler, resepter, leger og personer. I tillegg skal en lege ha en beholder over alle reseptene vedkommende har skrevet ut, og en person skal ha en beholder som inneholder alle personenes resepter. Legemidlene skal lagres i et objekt av klassen Tabell, reseptene skal lagres i et objekt av klassen EnkelReseptListe, legene skal lagres i et objekt av SortertEnkelListe og personene skal lagres i et objekt av klassen Tabell. Beholderen som inneholder en leges resepter skal være av klassen EldsteForstReseptListe, mens beholderen som inneholder en persons resepter skal være av klassen YngsteForstReseptListe.

Les gjennom hele oppgaven før du starter å svare på noen spørsmål. Du må ha klar oversikt over resten av oppgave 6 før du besvarer oppgavene 6.1.

Oppgave 6.1.a. Tegn opp klassehierarket for beholderne.
Ikke ta med navn på metoder og variable

Oppgave 6.1.b. Tegn opp datastrukturen.
Tegn alle beholderne, noen legemiddel-objekter, noen lege-objekter, noen person-objekter og noen resept-objekter. La det komme klart frem at en resept er med i mange beholdere. Tegn inn ”public” metodene (grensesnittet) i alle beholderene. Ikke ta med andre metoder. Når det er flere like objekter behøver du bare tegne inn all variable i ett objekt.

I oppgave 6.2 og 6.3.a frigjør vi oss fra leger, resepter mm. og definere generelle grensesnitt og klasser for beholdere:

Oppgave 6.2. Grensesnitt (interface) for beholdere

Skriv programmet for det generiske grensesnittet AbstraktTabell. Det skal ikke være noen restriksjoner på hva slags elementer den abstrakte tabellen skal kunne inneholde.
AbstraktTabell beskriver en beholder og du skal kunne:
- sette et objekt inn i tabellen på en oppgitt plass (indeks). Metoden returnerer sann eller usann avhengig om operasjonen gikk bra eller ikke.
- finn et objekt basert på en indeks.
- iterere over listen.

Skriv programmet for det generiske grensesnittet AbstraktSortertEnkelListe. En slik liste skal bare kunne inneholde elementer som implementerer grensesnittene Comparable (med seg selv) og Lik. En slik liste skal kunne:
- sette inn et nytt element (i sortert rekkefølge, minste først).
- finne et element basert på en nøkkel av typen String
- itereres over, slik at innholdet kan bli listet opp i sortert rekkefølge, minste først.

Oppgave 6.3. Klasser for beholdere

a) Generiske klasser

Skriv den generiske klassen Tabell som implementerer AbstraktTabell.
Klassen skal lagre alle elementene i en array, og arrayens lengde skal oppgis som parameter til konstruktøren.

Hvis du har lyst og tid: Når du setter noe inn i Tabellen og det ikke er plass, skal du lage en ny array som er lang nok (innenfor rimelighetens grenser), og så kopiere alle elementene over til den nye arrayen.

Skriv den generiske klassen SortertEnkelListe som implementerer AbstraktSortertEnkelListe som en enveisliste. Du skal programmere denne listen selv, ikke bruke klasser fra Java-biblioteket.

b) Ikke generiske klasser

Skriv klassen EnkelReseptListe. Klassen EnkelReseptListe skal inneholde en envisliste med en peker til første og en peker til siste element i listen. Klassen skal kunne ta vare på resepter, og en resept må kunne være med i flere objekter av denne klassen. Metodene i klassen skal kunne sette inn en resept og finne en resept basert på reseptnummeret. Hvis resepten som det letes etter ikke finnes i listen, skal det kastes et unntak. Skriv også en iterator over listen. Igjen skal du programmere denne listen selv, ikke bruke klasser fra Java-biblioteket.

Skriv subklassene EldsteForstReseptListe og YngsteForstReseptListe. Når du itererer over den første klassen skal du starte med den eldste resepten (den som ble satt inn først) og gå mot yngre (de som ble satt inn sist). Når du itererer i den andre klassen, skal du starte i den yngste enden.

Hint: Forskjellen på de to subklassene til klassen EnkelReseptListe kan være bare metoden som setter inn en resept.

Utfordring: I både SortertEnkelListe og EnkelReseptListe skal du skrive en iterator. Kan du klare å bruke den samme iteratoren i begge klassene? Da må de to listene kanskje ha de samme nodene?

Oppgave 6.4. Lag enkle enhetstester for alle beholderene.

Skriv et lite testprogram for hver av klassene Tabell, SortertEnkelListe, EldsteForstReseptListe og YngsteForstReseptListe. Prøv å lage programmene slik at både vanlige tilfeller og noen spesialtilfeller blir testet.

Innleveringen på denne oppgaven er tegningene fra oppgave 6.1, alle grensesnittene og klassene, og de fire kjørbare programmene fra oppgave 6.4