

EKSAMEN

6109

OBJEKTORIENTERT PROGRAMMERING

31.05.2016

Tid: 4 timer (9-13)

Målform: Bokmål

Sidetall: 5 (inkludert denne)

Hjelpemidler: Alle trykte og skrevne. Java API dokumentasjon er tilgjengelig lokalt på hver maskin.

Merknader: Eksamen blir gjennomført på PC uten tilgang på Internett, men satt opp med NetBeans og Sublime.

Besvarelsen skal lastes opp i WISEflow i form av et ferdig NetBeans-prosjekt pakket i en zip-fil.

Vedlegg: Ingen



Bokmål

Du skal opprette et NetBeans-prosjekt med navn `Eksamen` av type `JavaFX Application` med pakkenavn `eksamen`. Du skal også bruke samme navn på klasser og metoder som angitt i oppgaveteksten.

Noen deloppgaver kan medføre at du bør legge til kode i flere klasser. Du kan også legge til ekstra klasser og metoder hvis du synes det er hensiktsmessig. Pass på at koden er korrekt innrykket. Prosentstørrelse angitt for oppgaver og deloppgaver antyder vekt ved sensur. Prøv å svare på så mange oppgaver som mulig. Lykke til!

Innledning

Du skal lage deler av en kalender-applikasjon for lærere. En lærer kan ha to slags avtaler: veiledning med enkeltstudenter og forelesninger. Du skal etter hvert opprette klassene `Veiledning` og `Forelesning` for å representere slike avtaler. Om begge typer avtale skal man ta vare på starttidspunkt og varighet i antall timer, og dessuten noen opplysninger som gjelder hver av de to typene avtaler spesielt, som beskrevet under.

- `Veiledning`: Et objekt av denne klassen skal representere en veiledningsavtale med en enkelt student. Klassen bør ha en objektvariabel for å ta vare på studentens navn.
- `Forelesning`: Et objekt av denne klassen skal representere en "forelesningsrekke", det vil si en avtale som gjentar seg på et bestemt klokkeslett en bestemt ukedag hver uke. Det er altså alltid 7 dager mellom hver forelesning. Klassen bør ha objektvariable for å ta vare på emnenavn (en tekst) og antall repetisjoner (det vil si antall uker).

Klassene skal bruke en hjelpeklasse `Tidspunkt` for å ta vare på starttidspunkter. Alle avtaler for en bestemt lærer vil bli håndtert av en klasse `Kalender`. Du skal til slutt lage en `JavaFX`-applikasjon som gjør det mulig å søke i kalendrene til alle lærerne.

Oppgave 1 (50 %)

1-a (20 %)

Du skal først lage klassen `Tidspunkt` for å representere tidspunkter, dvs. en dato og et klokkeslett. Alle tidspunkter er hele timer, vi ser bort fra skuddår, og vi velger en litt spesiell representasjon i form av følgende tre objektvariable:

```
private int aar; // Årstallet
private int dagNr; // Dagene i et år fra 1 til 364
private int timeNr; // Timenummer fra 0 til 23
```

Det betyr at f.eks. tidspunktet kl. 11:00 på dato 03.02.2016 vil bli representert ved `aar=2016`, `dagNr=34` (31 dager i januar + 3 dager) og `timeNr=11`.

Man skal kunne lage nye `Tidspunkt`-objekter på to måter (samme eksempel som over):



```
Tidspunkt t1 = new Tidspunkt(2016, 34, 11);  
Tidspunkt t2 = new Tidspunkt(2016, 2, 3, 11);
```

Klassen skal ha get-metoder `getAar`, `getMnd` og `getDag` som alle returnerer heltall. Merk at `t1.getMnd()` skal returnere 2 og at `t1.getDag()` skal returnere 3 (igjen med samme eksempeldata som før).

Klassen skal dessuten ha følgende metoder:

```
public Tidspunkt leggTil(int antTimer) {  
    /* Tidspunktet antTimer etter dette tidspunktet. */  
}  
  
public int differanse(Tidspunkt t) {  
    /* Avstand i antall timer fra dette tidspunktet til t. */  
}  
  
@Override  
public String toString() {  
    /* Viser dette tidspunktet på formen dd.mm.åååå tt:00 */  
}
```

Lag klassen `Tidspunkt` som beskrevet her.

1-b (20 %)

Lag et klassehierarki som er egnet for å håndtere avtalene beskrevet tidligere. Det betyr at du skal lage klassene `Veiledning` og `Forelesning`. Dessuten er det hensiktsmessig med en eller flere abstrakte klasser. For hver klasse skal du deklare nødvendige objektvariabler, konstruktører med hensiktsmessige parametre samt `toString`-metoder.

1-c (10 %)

Begge klassene `Veiledning` og `Forelesning` skal ha boolske metoder `gjelderPåDag` og `kolliderer`.

Metoden `gjelderPåDag` tar tre heltallsparametre `aar`, `mnd` og `dag`, og returnerer `true` hvis og bare hvis denne avtalen gjelder på datoen bestemt av parametrene.

Metoden `kolliderer` tar en parameter `a` som enten kan være av typen `Veiledning` eller av typen `Forelesning`, samt tre heltallsparametre `aar`, `mnd` og `dag`, og returnerer `true` hvis og bare hvis denne avtalen kolliderer (overlapper) med parameteren `a` på datoen bestemt av parametrene `aar`, `mnd` og `dag`.

Tips: Et `Forelesning`-objekt som starter på dag nummer 34 i 2016 og som blir gjentatt 2 ganger (med 7 dagers mellomrom), vil også bli gjennomført på dag nummer 41 og 48.



Oppgave 2 (20 %)

Du skal nå lage en klasse `Kalender` som skal ta vare på alle avtalene til en lærer.

2-a (10 %)

Først skal du legge til rette for at klassene du laget i oppgave 1 kan brukes i en sortert `Collection`-klasse ved hjelp av grensesnittet `Comparable`.

Det er ønskelig å sortere avtaler kronologisk (sortert på starttid).

2-b (10 %)

Klassen `Kalender` skal ha objektvariabler for å ta vare på navnet til læreren (`String`), og en kronologisk sortert liste med avtaler (veiledningstimer og forelesninger). Listen skal representeres ved hjelp av klassen `TreeSet`.

Konstruktøren skal ta navnet på læreren som parameter. Listen med avtaler kan i utgangspunktet være tom.

Klassen skal ha en metode `nyAvtale` for å legge til enten en veiledningstime eller en forelesning.

Klassen skal også ha en metode `filter` som tar to tidspunkter som parametre, og som returnerer en sortert liste med alle avtaler mellom disse to tidspunktene.

Oppgave 3 (30 %)

Du skal nå lage et JavaFX brukergrensesnitt for visning av og søk i en samling av kalendre.

Kall GUI-klassen for `KalenderApp`.

Funksjonaliteten er beskrevet under. Du kan selv bestemme grafisk utforming.

3-a (5 %)

Lag en metode `lagTestData` som oppretter to kalendre (for to lærere) med minimum én veiledningstime og én forelesningsrekke i hver. Metoden skal opprette en objektsamling ved hjelp av en passende `Collection`-klasse, og returnere denne objektsamlingen.

3-b (25 %)

Lag GUI-klassen `KalenderApp`. Ved oppstart skal den bruke metoden `lagTestData` for å opprette testdata.



Brukergrensesnittet skal inneholde en tekstboks for å skrive inn navnet på en lærer, tekstbokser for å skrive inn to tidspunkter samt en knapp for å starte søket.

Som respons på søket skal programmet finne riktig kalender, filtrere ut de avtalene som ligger i intervallet mellom de to tidspunktene brukeren har skrevet inn, og til slutt vise disse i et tekstområde (hver avtale på en egen linje) – sortert kronologisk.