**Bunden forudsætning**

Adventure

Der skal laves et adventurespil hvor spilleren kan bevæge sig fra rum til rum, ved at skrive "go north", "go east", "go south" eller "go west" i konsollen. Spillet består således af nogle forbundne rum, alle med op til fire døre. Når en spiller er gået "east" for at gå ind i et rum, kan hun altid gå "west" for at komme tilbage igen.

Når spilleren er kommet ind i et nyt rum, skal programmet udskrive hvilket rum man nu befinder sig i, sammen med en lidt længere beskrivelse af omgivelserne. Det skal ikke udskrive hvilke retninger man kan bevæge sig i. Hvis spilleren skriver "go north" men der ikke er noget rum i den retning, altså ingen dør, skal programmet skrive "You cannot go that way", og forblive i det samme rum.

Krav

Spillet

Spillet skal opbygges af 9 rum - forbundet således:

A drawing of a grid with numbers and a compass

Description automatically generated

Spilleren starter altid i rum 1, øverst til venstre.

Bemærk at det er en "slags" labyrint, hvor det midterste rum (rum 5) er lidt sværere at komme til, og kun har én indgang - så måske er det rum noget særligt!

Rummene behøver ikke være faktiske "rum" i en bygning, men kan være grotter i en mine, eller områder i en skov, en borg, en rumstation, en fremmed planet - det er helt op til jer. Gør det spændende for spilleren at udforske jeres rum!

Brugerfladen

Brugerfladen skal være på engelsk - både kommandoer og beskrivelser af rum.  
  
Man skal kunne skrive "go north", "go east", "go south" og "go west" for at navigere rundt. Du må gerne forbedre brugerfladen, så man også kan nøjes med at skrive "north", "east", "south", "west", eller måske ligefrem blot "n", "e", "s", "w" - evt. kombineret med "go" - men den fulde version skal altid være mulig.  
  
Hvis man kan flytte sig i den ønskede retning, udskrives navn og beskrivelse for det nye rum man er i - hvis det ikke er muligt at flytte sig, får man beskeden "you cannot go that way" og forbliver hvor man var.

Man skal derudover kunne skrive disse kommandoer:

exit - for at afbryde spillet helt, og afslutte programmet

help - for at få en instruktion og oversigt over mulige kommandoer

look - for at få gentaget beskrivelsen af det rum man er i

Koden

Foruden **Main**klassen med main-metoden der starter det hele, skal du have mindst to klasser: **Adventure**som indeholder spillet - og **Room**som er et rum med et *navn*, en *beskrivelse*og *fire forbindelser* til andre rum.  
  
Hvis du allerede nu kan overskue at opdele yderligere, så put al brugerflade i en **UserInterface** klasse, men selve spillet (altså det at flytte sig fra rum til rum) er i Adventure. Hvis du hellere vil putte det hele i Adventure, så gør blot det til at starte med, men vær forberedt på, at det skal splittes op senere.

Room skal kende til sine naboer mod nord, syd, øst og vest og har derfor fire attributter af typen Room til at håndtere forbindelser til de fire andre rum: north, east, south og west. Hvis der ikke er noget rum i den pågældende retning, har attributten værdien null.  
  
Det er Adventure-klassen der skal bygge "kortet" ved at oprette og forbinde de ni room objekter. Det skal gøres ved at Adventure-klassen opretter rum objekterne og for hvert rum angiver (setter) rummets naboer mod nord, syd, øst og vest - hvis de har de naboer i de enkelte retning. Der skal ikke oprettes arrays eller andre datastrukturer med “kortet”.

Hvert room objekt har attributten name - det er en god ide at bruge samme værdi til denne attribut, som navnet på variablen du opretter rummet med i Adventure-klassen. Det vil gøre det langt lettere at debugge koden! Eksempel:

Room room1 = new Room("Room 1", "room with no distinct features, except two doors");  
  
Al input/output og kommunikation med brugeren skal foregå via UserInterface, hvis du har denne klasse, ellers i Adventure-objektet.

Adventure-objektet holder hele tiden styr på hvilket rum spilleren befinder sig i - f.eks. med en variabel: currentRoom.

Anbefalet procedure

1. Start med at lave brugerfladen - uden nogle rooms. Bare tag imod input, og fortolk det, så programmet for eksempel udskriver "going north" når brugeren har skrevet en af kommandoerne for at "go north", og "looking around", når brugeren har skrevet "look" og så fremdeles.
2. Lav derefter Room klassen, og opret et enkelt room-objekt i Adventure. Lad det være currentRoom, som spilleren befinder sig i. Udskriv navn og beskrivelse når spilleren skriver "look".
3. Opret de otte andre room-objekter, med navne og beskrivelser, og forbind dem - husk at når room1 skal være forbundet til room2 mod east, så skal room2 tilsvarende være forbundet til room1 mod west! Igen, husk at koden til at oprette disse forbindelser skal ligge i Adventure-objektet!
4. Lav koden til at navigere rundt imellem rum - tænk på det som at currentRoom er der hvor spilleren p.t. befinder sig. Når der bliver skrevet "go north", så beder spilleren om at blive flyttet til det room der er north for currentRoom - og hvis det room findes, så er alt godt, og currentRoom er nu det nye requested room - men hvis der ikke var noget room, skal spilleren have "you cannot go that way" beskeden.

Aflevering

Hele Adventure-projektet er én **bunden forudsætning** – det vil sige at du **skal**aflevere alle dele af projektet, for at blive indstillet til eksamen.

Hvordan

Opret ét GitHub-repository og sørg for at al source-koden er committed og pushed før deadline.

Aflever linket til repositoriet – ikke til den enkelte fil eller mappe, men til repositoriet som et hele.

Hvornår

Inden fredag, da I skal bruge tiden her på refaktorering af jeres arbejde med Adventure del 1.