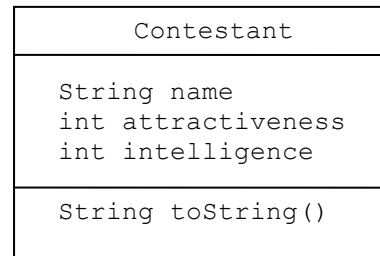


## Eksamensprojekt 6

1. Opret en klasse, *Contestant*, der repræsenterer en deltager i Paradise Hotel; klassen *Contestant* er specificeret i UML-diagrammet til højre. De tre feltvariabler skal initialiseres i en konstruktør (via parametre af passende type). Metoden *toString* skal returnere en streng-repræsentation for en deltager, f.eks.

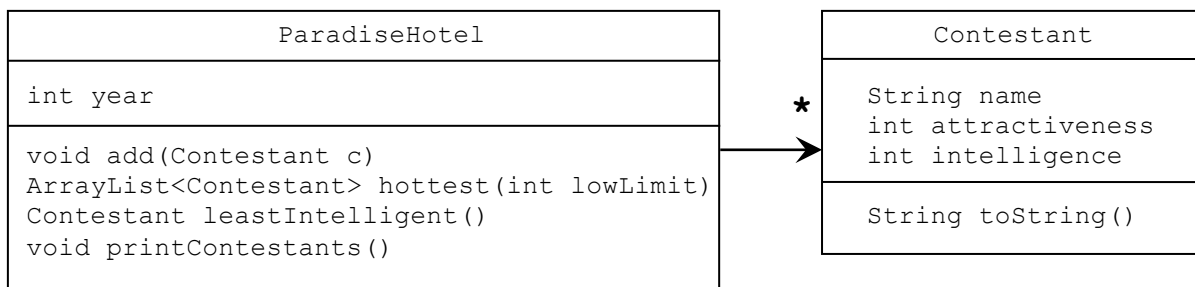
`"Amalie, attractiveness: 7, intelligence: 0"`



2. Lav en *Driver*-klasse med en *exam*-metode. Metoden skal være static, have returtype void og være uden parametre.
3. Opret fem velvalgte *Contestant*-objekter, via objektreferencer *c1*, *c2*, *c3*, *c4* og *c5* i *exam*-metoden og udskriv disse vha. *toString*-metoden.

**Tilkald tilsynsførende og demonstrer det du har lavet indtil nu.**

4. Opret en ny klasse, *ParadiseHotel*, der repræsenterer en udgave med et antal deltagere. Klassen *ParadiseHotel*, og dens relation til klassen *Contestant*, er specificeret i følgende UML-diagram:



5. Programmér metoden *add* der tilføjer *Contestant*-objektet *c* til *ParadiseHotel*-objektet.
6. Opret et objekt af typen *ParadiseHotel* i *exam*-metoden i *Driver*-klassen og knyt de allerede oprettede *Contestant*-objekter hertil.
7. Programmér metoden *hottest*. Metoden skal returnere en liste med de deltagere der er mere attraktive (*attractiveness*) end *lowLimit*. Udvid *Contestant*-klassen med de nødvendige get-metoder.
8. Benyt metoden *hottest* til i *exam*-metoden i *Driver*-klassen at udskrive information om de deltagere, der er mere attraktive end en vis en given værdi (vælg selv).

**Tilkald tilsynsførende og demonstrer det du har lavet indtil nu.**

9. Programmér metoden *leastIntelligent*. Metoden skal returnere den deltager der har lavest intelligens (*intelligence*). Udvid *Contestant*-klassen med de nødvendige get-metoder. Aftest metoden fra *Driver*-klassen.
10. Programmér metoden *printContestants*. Metoden skal udskrive en liste af samtlige deltagere på denne udgave af Paradise Hotel, ordnet efter *intelligence*. Hvis to deltagere har samme *intelligence*, skal de ordnes alfabetisk efter deres navn (*name*). Aftest metoden fra *Driver*-klassen.

**Tilkald tilsynsførende og demonstrer din færdige løsning.**