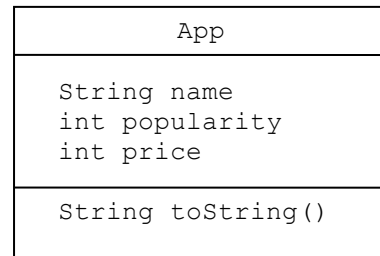


Eksamensprojekt 2

1. Opret en klasse, *App*, der repræsenterer en telefonapplikation; klassen *App* 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 *App*, f.eks.

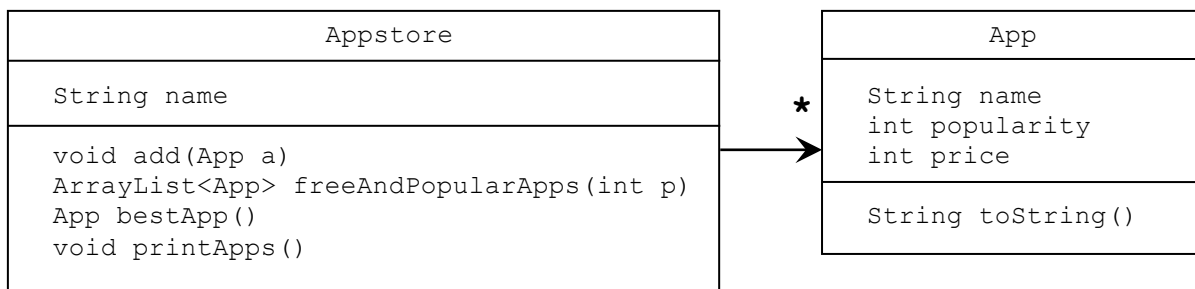
"Angry Birds, 100000 downloads, 10 usd"



2. Lav en *Driver*-klasse med en *exam*-metode. Metoden skal være static, have returtype void og være uden parametre.
3. Opret fem velvalgte *App*-objekter, via objektreferencer *a1*, *a2*, *a3*, *a4* og *a5* i *exam*-metoden og udskriv disse vha. *toString*-metoden.

Tilkald tilsynsførende og demonstrer det du har lavet indtil nu.

4. Opret en ny klasse, *Appstore*, der repræsenterer en butik med et antal *Apps*. Klassen *Appstore*, og dens relation til klassen *App*, er specificeret i følgende UML-diagram:



5. Programmér metoden *add* der tilføjer *App*-objektet *a* til *Appstore*-objektet.
6. Opret et objekt af typen *Appstore* i *exam*-metoden i *Driver*-klassen og knyt de allerede oprettede *App*-objekter hertil.
7. Programmér metoden *freeAndPopularApps*. Metoden skal returnere en liste med de *Apps* der er gratis og har mere end *p* downloads. Udvid *App*-klassen med de nødvendige get-metoder.
8. Benyt metoden *freeAndPopularApps* til i *exam*-metoden i *Driver*-klassen at udskrive information om de *Apps* der er populære og gratis.

Tilkald tilsynsførende og demonstrer det du har lavet indtil nu.

9. Programmér metoden *bestApp*. Metoden skal returnere den *App* der har flest downloads. Aftest metoden fra *Driver*-klassen.
10. Programmér metoden *printApps*. Metoden skal udskrive en liste af samtlige *Apps* i en *Appstore*, ordnet efter popularitet (*Apps* med samme popularitet skal ordnes alfabetisk efter navn). Aftest metoden fra *Driver*-klassen.

Tilkald tilsynsførende og demonstrer din færdige løsning.