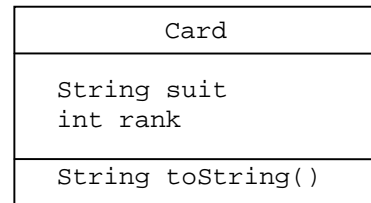


Eksamensprojekt 2.2.4 (1)

1. Opret en klasse *Card*, der repræsenterer et spillekort; klassen *Card* er specificeret i UML-diagrammet til højre (*suit* kan antage værdierne "spade", "heart", "club", eller "diamond", og *rank* kan antage en af værdierne 1, ..., 13). De to feltvariabler skal initialiseres i en konstruktør (via parametre af passende type). Metoden *toString* skal returnere en streng-repræsentation for et kort, f.eks.

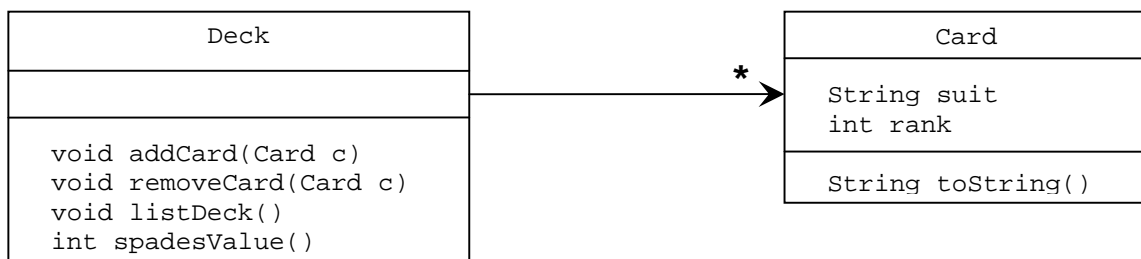


"heart 12"

2. Lav en ny testmetode i *Driver*-klassen, og giv metoden navnet *exam*. Metoden skal være static, have returtype void og ingen parametre.
3. Opret to *Card*-objekter, via objektreferencer *c1* og *c2*, i *exam*-metoden og udskriv disse vha. *toString*-metoden.

Tilkald tilsynsførende til gennemsyn af det du har lavet indtil nu; klargør fremvisningen ved at åbne de relevante filer og placere dem synligt ved siden af hinanden på skærmen.

4. Opret en ny klasse, *Deck*, der repræsenterer en samling kort (f.eks. en bridgehånd eller et spil kort); klassen *Deck*, og dens relation til klassen *Card*, er specificeret i dette UML-diagram:



5. Programmér metoderne *addCard* og *removeCard* der henholdsvis tilføjer og fjerner *Card*-objektet *c* til/fra *Deck*-objektet.
6. Opret et objekt af typen *Deck* i *exam*-metoden i *Driver*-klassen og knyt de to allerede oprettede *Card*-objekter hertil.
7. Programmér metoden *listDeck*; metoden skal udskrive en liste over de kort den består af (et kort pr. linie). Test metoden med en passende udvidelse af *exam*-metoden i *Driver*-klassen.
8. Lad klassen *Card* implementere interfacet *Comparable* idet ordningen på *Card*-objekter defineres af værdien af *rank*; hvis to kort har samme *rank*, er det *suit* der afgør hvilket der er størst (Spar > Hjerter > Klør > Ruder). Test implementeringen af *Comparable* med en passende udvidelse af *exam*-metoden i *Driver*-klassen.