- Crea un projecte *BlueJ* anomenat exercicis en la teua carpeta IIP on implementar les classes proposades en aquest document.

Exercicis - IIP Primer Parcial

1. 7 punts Es desitja fer una aplicació per gestionar un catàleg d'estels. Es demana implementar la classe `Astre` i per a això s'ha de:

   Tema 2: crea la classe tipus de dades Astre en el projecte exercicis

   Tema 3
   a) (0.5 punts) Definir els atributs de classe públics i constants per tal de representar els diferents tipus d'astres considerats, `ESTEL`, `NEBULOSA` i `GALAXIA`, amb valors 0, 1 i 2, respectivament. S'han d'utilitzar sempre que es requerisca (tant a la classe `Astre` com a la classe `TestAstre`).

   b) (0.5 punts) Definir els atributs d'instància privats `nom` (`String`), `tipus` (`int`), `brillantor` (`double` que representa la seua brillantor aparent) i `distancia` en anys llum (`double`).

   Tema 4
   c) (1 punt) Implementar dos constructors:
   - Un constructor general amb els paràmetres apropiats per a inicialitzar tots els atributs d'instància; es pot suposar que tots els paràmetres tenen valors correctes.
   - Un constructor per defecte que cree un `Astre` de nom "Sirius", de tipus `ESTEL`, brillantor -1.42 i distància 8.7.

   d) (0.5 punts) Escriure el mètode consultor i el mètode modificador de l'atribut `brillantor`.

   e) (1 punt) Escriure el mètode `equals` (que sobreescriu el d'`Object`) per a comprovar si dos astres són iguals. Dos astres són iguals si tots els seus atributs coincideixen.

   Tema 5
   f) (1 punt) Escriure el mètode `toString` (que sobreescriu el d'`Object`) per a que torne un `String` amb el següent format: "`nom`: `tipus` (`brillantor`, `distancia`)"; p.e., "Sirius: Estel (-1.42, 8.70)". Tots els números han d'arredonir-se a només dues xifres decimals i el tipus de l'astre ha d'apareixer com "Estel", "Nebulosa" o "Galàxia". S'ha d'utilitzar la instrucció `switch`.

   Tema 4
   g) (0.5 punts) Escriure el mètode `magnitudAbsoluta` que calcule la magnitud absoluta (la brillantor que tindria si l'astre estiguera situat a una distància fixa) aplicant la següent fórmula: $M = b + 5 \log d$, sent M la magnitud absoluta, b la brillantor aparent i d la distància. Usar el mètode `log10` de la llibreria `Math` per tal de calcular el logaritme de la fórmula.

   Tema 3

   Tema 5
   h) (1 punt) Escriure el mètode `mesBrillant` que torne 1 si l'`Astre` actual és més brillant en magnitud absoluta que un `Astre` donat, 0 si tenen la mateixa magnitud absoluta i -1 si el `Astre` donat és més brillant en magnitud absoluta que l'actual. Cal notar que s'usa la magnitud absoluta dels dos astres ja que, només situant-los a la mateixa distància, pot comparar-se la seua lluminositat.

   i) (1 punt) Escriure el mètode `visibleAmb` que torne un `String` que descriga la forma en la que l'astre pot ser observat, tenint en compte el següent:

Brillantor	Visibilitat
< 5	"a simple vista"
≥ 5 i < 7	"amb prismàtics"
≥ 7 i ≤ 25	"amb telescopi"
> 25	"amb grans telescopis"

2. 3 punts Utilitzant la classe desenvolupada en l'exercici anterior, es demana implementar la classe `TestAstre` amb un mètode `main` que realitze les següents accions:

   Tema 2: crea la classe programa TestAstre en el projecte exercicis

   Tema 4
   a) Crear un objecte de tipus `Astre` per a l'estel "Alfa Centauri" que té una brillantor 4.6 i està a una distància de 4.3 anys llum. A continuació mostrar les seues dades per pantalla.

   Tema 3
   b) Crear un objecte de tipus `Astre` preguntant a l'usuari el nom, tipus, brillantor i distància. Mostrar per pantalla com es pot observar aquest astre.

   Tema 5
   c) Mostrar un missatge per pantalla indicant quin és l'`Astre` més brillant en magnitud absoluta dels dos que s'han creat.

3. 6 punts S'està desenvolupant una aplicació que usa una baralla de cartes. Per a això, s'implementaran en Java les classes necessaries. Una d'elles és la classe `Carta` que permet representar una carta de la baralla espanyola. La informació requerida per tal d'identificar una `Carta` és:

- el seu pal (oros, copes, espases o bastos) i
- el seu valor (un enter entre 1 i 12).

Per a aquesta classe, **es demana**:

Tema 2: crea la classe tipus de dades Carta en el projecte exercicis

Tema 3
a) (0.5 punts) Definir 4 constants, atributs de classe (estàtics) públics enters, per a representar cadascun dels pals de la baralla (`OROS` serà el valor 0, `COPES` l'1, `ESPASES` el 2 i `BASTOS` el 3).

b) (0.5 punts) Definir els atributs d'instància de la classe: `pal` i `valor`. Tema 3

Tema 4
c) (1 punt) Escriure dos constructors: un per a construir una carta de forma aleatòria i un altre per a construir una carta d'acord a dues dades: el seu pal i el seu valor. Suposem que aquestes dades són correctes.

d) (0.5 punts) Escriure dos mètodes consultors i dos mètodes modificadors dels valors dels atributs d'instància.

Tema 5
e) (1 punt) Escriure un mètode `esMenor` per a comprovar si la carta actual és menor que un altra carta donada. El criteri d'ordenació és per pals (el menor és ors, després copes, a continuació espases i, finalment, bastos) i dins de cada pal per valor (1, 2, .., 12).

Tema 4
f) (0.5 punts) Escriure un mètode `segPal` per a tornar una nova carta amb el mateix valor que el de la carta actual però del pal següent, segons l'ordenació anterior i sabent que el següent al pal de bastos és oros. Tema 3

g) (1 punt) Escriure un mètode `equals` per a comprovar la igualtat de dues cartes (sobreescriptura del mètode `equals` d'`Object`).

Tema 5
h) (1 punt) Escriure un mètode `toString` per a transformar en `String` la carta actual, amb el següent format: "valor de pal"; per exemple, "4 d'oros" o "1 de bastos" (sobreescriptura del mètode `toString` d'`Object`).

4. 4 punts Considerant la classe anterior, **es demana**: implementar en Java una classe `JocCartes` amb el mètodes següents:

Tema 2: crea la classe programa JocCartes en el projecte exercicis

Tema 5
a) (2 punts) Un mètode de classe (estàtic) que donats dos objectes `Carta` i un número enter representant el pal de triomf (o pal guanyador), determine quina és la carta guanyadora. El mètode ha de tornar 0 si les dues cartes són iguals. En un altre cas, tornarà -1 quan la primera carta és la guanyadora i 1 quan la segona carta és la guanyadora.

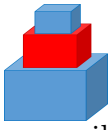
Per a determinar la carta guanyadora s'aplicaran les següents regles:

- Si les dues cartes són del mateix pal, la carta guanyadora és l'as (valor 1) i, en la resta de casos, la carta guanyadora és la de valor més alt (per exemple, "1 d'oros" guanya a "7 d'oros", "5 de copes" guanya a "2 de copes", "11 de bastos" guanya a "7 de bastos").
- Si les dues cartes són de pals diferents:
  - Si el pal d'alguna carta és el pal de triomf, eixa carta és la guanyadora.
  - En un altre cas, la primera carta sempre guanya a la segona.

b) (2 punts) Un mètode `main` on obligatòriament s'ha de: Tema 3

Tema 4
1. Crear una `Carta` a partir d'un pal i un valor donats (sol.licitats a l'usuari des de teclat), i mostrar les seues dades per pantalla.
2. Crear una `Carta` aleatòriament i mostrar les seues dades per pantalla. Tema 3

Tema 5
3. Generar aleatòriament un enter en el rang [0..3] representant el pal de triomf, i mostrar per pantalla a quin pal correspon.
4. Mostrar per pantalla les dades de la carta guanyadora (invocant al mètode de l'apartat anterior amb l'objecte `Carta` de l'usuari com a primer argument).



Es desitja fer una aplicació per a representar un joc amb blocs de diferents colors i dimensions, apilables en torres. Cadascun d'aquests blocs té associat un **color** (blau o roig) i una **dimensió** (nombre enter entre 1 i 50, tots dos inclosos).

Les regles d'aquest joc indiquen que:

- Els blocs apilats en una torre han de seguir colors alterns (damunt d'un bloc blau solament pot haver-hi un bloc roig i viceversa).
- Damunt d'un bloc de dimensió x solament pot haver-hi un bloc de dimensió y, on $y \leq x$ (la torre s'estreny cap a la punta, és a dir, s'eixampla cap a la base).
- Un bloc pot ser un **comodí**, en aquest cas pot anar damunt de qualsevol altre bloc, independentment dels seus colors. Ara bé, un bloc comodí, com qualsevol altre, ha de respectar la regla de la dimensió.

5. 6 punts Es demana implementar la classe `Bloc`, per al que s'ha de: Tema 2: crea la classe tipus de dades Bloc en el projecte exercicis

Tema 3
a) (0.5 punts) Definir els atributs de classe públics i constants que representen els dos colors possibles: `BLAU` i `ROIG`, amb valors enters 0 i 1, respectivament. S'han d'utilitzar sempre que es requerisca (tant en la classe `Bloc` com en la classe `TorreBlocs`).

b) (0.5 punts) Definir els atributs d'instància privats `color` (`int`), `dimensio` (`int`), `comodi` (`boolean`).

Tema 4
c) (1.5 punts) Implementar dos constructors:
- Un constructor general amb els paràmetres apropiats per a inicialitzar tots els atributs d'instància.
- Un constructor per defecte que cree un `Bloc` blau que no siga un comodí i la dimensió del qual es determine aleatòriament dins del rang [1,50]. Tema 3

d) (0.5 punts) Escriure el mètode consultor i el mètode modificador de l'atribut `dimensio`.

e) (1 punt) Escriure el mètode `equals` (que sobreescriu el d'`Object`) per a comprovar si un bloc és igual a un altre, açò és, si l'altre és un bloc i els atributs d'un i l'altre coincideixen un a un.

Tema 5
f) (1 punt) Escriure el mètode `toString` (que sobreescriu el d'`Object`) perquè el resultat tinga un format com el mostrat en els següents exemples:
"`(Color: roig, dimensió: 22 i SÍ és comodí)`",
"`(Color: blau, dimensió: 15 i NO és comodí)`".

g) (1 punt) Escriure un mètode `potEstarDamuntDe(Bloc b)` que comprove si un bloc, aplicant les regles ja esmentades, pot estar o no damunt d'un altre `Bloc b` que reb com a argument. Per exemple, donades dues variables `a` i `b` de tipus `Bloc`, `a.potEstarDamuntDe(b)` serà cert si i solament si la dimensió del bloc `a` és menor o igual que la del bloc `b` i, bé `a` és un comodí, o bé els colors de `a` i `b` són diferents.

6. 4 punts Es demana implementar la classe programa `TorreBlocs`, l'objectiu de la qual és poder fer proves amb torres menudes d'alguns blocs. Per a açò, cal implementar aquesta classe amb un mètode `dimensioValida` que: Tema 2: crea la classe programa TorreBlocs en el projecte exercicis

Tema 5
a) (1.25 punts) limite el valor de la dimensió `d` que se li passa com a paràmetre a un dels valors de l'interval [1,50]. Açò significa que si `d` és menor que 1, torna 1; si `d` és major que 50, torna 50; i, en qualsevol altre cas, torna `d` sense modificar-lo.

i amb un mètode `main` que realitze les següents accions:

Tema 4
b) (0.25 punts) Crear un `Bloc b1` amb el constructor per defecte.

c) (0.5 punts) Crear un `Bloc b2` de color blau, dimensió 30 i que siga un comodí (amb el constructor general).

Tema 3

d) (1 punt) Després de llegir el seu color i dimensió des de teclat, crear un `Bloc b3` que no siga comodí.

Tema 4

El color se sol·licitarà a l'usuari com un `String`, amb valors "roig" o "blau"; si l'usuari introdueix qualsevol altre valor, el color del `Bloc` serà roig.

El valor de dimensió se sol·licitarà a l'usuari com un enter en el rang [1,50] i, per tal de garantizar que l'usuari no ha introduït qualsevol altre valor, s'haurà d'invocar al mètode `dimensioValida`.

e) (0.25 punts) Mostrar per pantalla els tres objectes creats.

Tema 5

f) (0.75 punts) Comprovar si es pot formar una torre situant el bloc `b3` damunt del `b2` i aquest últim damunt del bloc `b1`. A continuació, mostrar per pantalla un missatge amb el resultat d'aquesta comprovació, açò és, si s'ha pogut formar la torre o no.

S'han d'importar les classes que es consideren necessàries i utilitzar les constants definides en la classe `Bloc` on siga oportú.