Programació orientada a objectes

## Exercici 1 – Herència i classes estàtiques

1. Crearem pas per pas una simulació de la estructura d’una empresa:

* Crea la classe Treballador amb els següents atributs i mètodes:
  + Nom
  + Direcció
  + Data Incorporació
  + Sou Base
  + IRPF
* calculaSouMensualNet()
* Crea la classe TreballadorInformàtic amb els següents atributs i mètodes:
  + Nom
  + Direcció
  + Data Incorporació
  + Sou Base
  + IRPF
  + EquipInformàtic
* calculaSouMensualNet()
* Crea la classe Repartidor amb els següents atributs i mètodes:
  + Nom
  + Direcció
  + Data Incorporació
  + Sou Base
  + IRPF
  + Mòbil
* calculaSouMensualNet()

Com pots intuir, és redundant la creació de les classes repetint pràcticament tots els atributs i mètodes. A continuació aplicarem herència, creant la resta de classes a partir de la classe treballador. (De moment EquipInformàtic i Mòvil son un string amb el número de sèrie).

1. Crea tres instàncies de cada classe i mostra en php una taula per cada classe de treballador que t’indiqui les seves dades, amb una fila final que t’indicarà el sou anualBrutTotal i el souMensualBrutTotal per treballador. (Com a mínim un treballador per classe de sou inferior a 17000)

Fes ús d’una classe estàtica auxiliar calculadoraSous amb un mètode que et calculi la suma dels sous dels treballadors que li passes per paràmetre.

1. Degut la situació de crisis actual volem analitzar la possibilitat d’incrementar el souBase dels nostres treballadors que cobrin menys de 17000 un 10% en els propers mesos. Afegeix una columna més a les taules que em marqui en vermell el sou actualitzat i a la fila final em calculi que m’implicarà aquest increment. (Utiliza la classe estàtica auxiliar)

## Exercici 2 – Abstracció

1. Crearem una classe nova anomenada Gerents, amb els següents atributs i mètodes:
   * Nom
   * Direcció
   * Data Incorporació
   * Càrrec
   * Complement
   * Sou Base
   * IRPF

* calculaSouMensualNet()

Podríem heredar de nou, però el mètode calculaSouMensualNet ha de tenir en compte el complement. Per aquesta raó crearem una classe abstracta “Personal” amb els següents atributs:

* + Nom
  + Direcció
  + Data Incorporació
  + EquipInformàtic
  + Sou Base
  + IRPF
* Abstract function calculaSouMensualNet()

La nova classe gerent i l’anterior classe treballador implementaran la classe abstracta Personal.

Actualitza les classes, crea tres gerents i torna a mostra les taules anteriors.

1. Desprès d’analitzar en el exercici anterior una modificació de sous, la volem aplicar el proper mes. Dintre de la classe Personal crea un mètode abstracte souMensualActualitzat, que s’implementarà de la següent manera:
   1. Gerents: reducció del 50% del complement
   2. Treballadors: increment del 10% del sou mensual si és inferior a 17000. A més, tots els repartidors tindran un increment del sou mensual del 8% degut al preu del combustible.

La classe repartidor haurà de sobreescriure el mètode souMensualActualitzat heretat de la classe Treballador.

Mostra les taules anteriors, marcant en vermell en l´última columna el sou mensuals actualitzats.

1. Crea una classe abstracta Dispositiu amb atributs: numSerie, marca, model, preu, anyCompra, propietari (defecte null) i mètode abstracte calculaValor.

Crea les subClasses PC i Mòbil, la classe mòbil tindrà l’atribut número i el PC sobretaula (True or False). El mètode calcula valor en el cas dels mòbils reduirà en un 25% el seu preuInicial per cada any i en el cas dels pc un 15% en els portàtils i un 10% en els del sobretaula.

Mostra una taula amb l’inventari, indicant el valor actual de cada equip.

1. Crea de nou les classes treballador tenint en compte les noves classes Creades ( els gerents tenen portàtils i els informàtics equips de sorbetaula).