**package alumnos;**

**public interface Calificacion {**

**public double calificacion\_parcial();**

**public double calificacion\_final();**

**}**

**package alumnos;**

**public abstract class Alumno implements Calificacion {**

**//atributos**

**protected String nombre;**

**protected String apellidos;**

**protected String dni;**

**protected String fecha\_nacimiento;**

**protected String domicilio;**

**protected String curso;**

**protected int escalafon;**

**protected Asignatura asignaturas[];**

**//constructores**

**public Alumno(String nombre, String apellidos){**

**this.nombre=nombre;**

**this.apellidos=apellidos;**

**dni=domicilio=fecha\_nacimiento="";**

**}**

**//métodos set/get**

**protected void setNombre(String nombre){**

**this.nombre=nombre;**

**}**

**protected String getNombre(){ return nombre;**

**}**

**protected void setApellidos(String apellidos){**

**this.apellidos=apellidos;**

**}**

**protected String getApellidos(){ return apellidos;**

**}**

**protected String getDni(){**

**return dni;**

**}**

**protected String setDni(int numero){**

**String nif;**

**nif=Extras.*nif*(numero);**

**return nif;**

**}**

**protected String getDomicilio(){**

**return domicilio;**

**}**

**protected char setFechaNacimiento(String fecha){**

**//llama al método que comprueba si es correcta la fecha**

**//si es correcta, la guarda en el atributo fecha\_nacimiento y devuelve la letra ‘t’**

**//si no es correcta devuelve 'f' y que vuelva a pedir la fecha**

**boolean correcta=false;**

**correcta=comprueba\_fecha(fecha);**

**if (correcta){**

**fecha\_nacimiento=fecha;**

**return 't';**

**}else return 'f';**

**}**

**protected String getFechaNacimiento(){**

**return fecha\_nacimiento;**

**}**

**protected void setEscalafon(int orden){**

**escalafon=orden;**

**}**

**protected int getEscalafon(){**

**return escalafon;**

**}**

**public void setCurso(String curse){**

**curso=curse;**

**}**

**public String getCurso(){**

**return curso;**

**}**

**//resto de métodos**

**protected int edad(){**

**int edad=0;**

**//Calcular la edad por la fecha de nacimiento**

**return edad;**

**}**

**protected boolean comprueba\_fecha(String fecha){**

**boolean fecha\_correcta=false;**

**//método que compruebe si fecha tiene el formato "dd/mm/aaaa"**

**if (fecha\_correcta) return true;**

**else return false;**

**}**

**protected void rellena\_asignaturas(){**

**//método que rellene automáticamente el array de asignaturas**

**//para no tener que ir rellenandolas a mano**

**Asignatura a1 = new Asignatura("Java",8);**

**asignaturas[0]=a1;**

**//el nombre de las asignaturas con el método estatico Extras.rellena\_cadena**

**//la nota de la asignatura con el método estatico Extras.genera\_numero**

**}**

**//métodos abstractos**

**protected abstract void setDomicilio(String domicilio);**

**//método abstracto para guardar el domicilio según el alumno derivado**

**protected abstract String visualizar\_alumno();**

**}**

**Resumiendo, en esta clase Alumno, tendrá que implementar (o acabar) los siguientes métodos:**

**protected char setFechaNacimiento(String fecha){**

**protected int edad(){**

**protected boolean comprueba\_fecha(String fecha){**

**protected void rellena\_asignaturas(){**

**package alumnos;**

**import java.util.Scanner;**

**public class Extras {**

**private static Scanner *tc* = new Scanner(System.*in*);**

**//métodos staticos**

**public static String nif(int dni){**

**//Tendrá que usar obligatoriamiente la cadena letras en la obtención de la letra del nif**

**//usando algún método de manejo de cadenas como toCharArray(), charAt()**

**String letras= "TRWAGMYFPDXBNJZSQVHLCKE";**

**String nif="";**

**//metodo que calcula la letra del dni pasado y devolvemos el nif**

**return nif;**

**}**

**public static String rellena\_cadena(int tamanyo){**

**String salida="";**

**//um método que cree una cadena de letras aleatorias del tamanyo indicado**

**return salida;**

**}**

**public static double numero\_aleatorio(int inicio,int fin){**

**double generado;**

**generado=((Math.*random*()\*fin)+inicio);**

**return generado;**

**}**

**public static int menu(){**

**int opcion=0;**

**System.*out*.println("\n1.- Alta Alumno Formacion");**

**System.*out*.println("2.- Alta Alumno Perfeccionamiento");**

**System.*out*.println("3.- Baja Alumno Formacion");**

**System.*out*.println("4.- Baja Alumno Perfeccionamiento");**

**System.*out*.println("5.- Sanciona Alumno Formacion");**

**System.*out*.println("6.- Ausencia Alumno Perfeccinamiento");**

**System.*out*.println("7.- Consulta Sanciones Alumnos Formacion");**

**System.*out*.println("8.- Consulta Ausencias Alumnos Perfeccionamiento");**

**System.*out*.println("9.- Consulta Calificaciones Finales Alumnos");**

**System.*out*.println("0.- Salir");**

**System.*out*.print(">>> ");**

**opcion=*tc*.nextInt();**

**return opcion;**

**}**

**}**

**Resumiendo, en esta clase Extras, tendrá que implementar (o acabar) los siguientes métodos:**

**public static String nif(int dni)**

**public static String rellena\_cadena(int tamanyo)**

**package alumnos;**

**public interface Ipa {**

**public double ipa();**

**}**

**package alumnos;**

**public class Asignatura {**

**private String asignatura;**

**private int nota;**

**public Asignatura (String asignatura, int nota){**

**this.asignatura=asignatura;**

**this.nota=nota;**

**}**

**public String mostrar\_asignatura(){**

**return asignatura+" : " + nota;**

**}**

**}**

**package alumnos;**

**public class Ausencias {**

**private String fecha;**

**private int horas;**

**public Ausencias(String fecha, int horas){**

**this.fecha=fecha;**

**this.horas=horas;**

**}**

**//get necesario**

**public int getHoras(){**

**return horas;**

**}**

**public String mostrar\_ausencias(){**

**return "Día " + fecha + ", " + horas + " horas";**

**}**

**}**

**package alumnos;**

**public class Sancion {**

**private String tipo\_sancion;**

**private int dias\_sancion;**

**public Sancion(String tipo, int dias){**

**tipo\_sancion=tipo;**

**dias\_sancion=dias;**

**}**

**public String mostrar\_sancion(){**

**return dias\_sancion + "dias de " + tipo\_sancion;**

**}**

**}**

**package alumnos;**

**public abstract class AlumnoFormacion extends Alumno implements Ipa {**

**//atributos**

**protected Sancion san[];**

**protected int cantidad=0;**

**//constructor**

**public AlumnoFormacion(String nombre, String apellidos, String curso){**

**super(nombre,apellidos);**

**san = new Sancion[10];**

**this.curso=curso;**

**//para dar valores a dni y fecha\_nacimiento, usar métodos de la clase padre**

**setDomicilio("Emacot"); }**

**//getCantidad necesario para devolver la cantidad de sanciones**

**public int getCantidad(){**

**return cantidad; }**

**//resto métodos**

**public void setDomicilio(String Domicilio){**

**domicilio=Domicilio;**

**}**

**public void anyadir\_sancion(String tipo, int dias){**

**//añadir un objeto sancion al array de sanciones**

**//usar el atributo cantidad para controlar el máximo de 10 sanciones**

**}**

**public void mostrar\_sanciones\_alumno(){**

**//hay que usar el método mostrar\_sancion en todos**

**//los objetos sancion del array**

**}**

**public String visualizar\_alumno(){**

**return "<AlumnoFormacion>" + this.toString();**

**}**

**}**

**Resumiendo, en esta clase AlumnoFormación, tendrá que implementar los siguientes métodos:**

**public void anyadir\_sancion(String tipo, int dias)**

**public void mostrar\_sanciones\_alumno()**

**package alumnos;**

**public abstract class AlumnoPerfeccionamiento extends Alumno {**

**//atributos**

**protected Ausencias aus[];**

**protected int cantidad=0;**

**protected int acumuladas=0;**

**//constructor**

**public AlumnoPerfeccionamiento(String nombre, String apellidos, String curso){**

**super(nombre,apellidos);**

**aus = new Ausencias[10];**

**this.curso=curso;**

**//para dar valores a dni y fecha\_nacimiento, usar métodos de la clase padre**

**}**

**//get/set necesarios**

**public int getCantidad(){**

**return cantidad;**

**}**

**public int getAcumuludas(){**

**return acumuladas;**

**}**

**public void setDomicilio(String home){**

**domicilio=home;**

**}**

**public void anyadir\_ausencia(String fecha, int horas){**

**//añadir un objeto sancion al array de sanciones**

**//Ausencias a = new Ausencias(fecha,horas)**

**//aus[cantidad]=a;**

**//usar el atributo cantidad para controlar el máximo de 10 sanciones**

**//iremos sumando en acumuladas la cantidad total de horas**

**//acumuladas=acumuladas+a.getHoras();**

**}**

**public void mostrar\_ausencias\_alumno(){**

**//hay que usar el método mostrar\_ausencia en todos**

**//los objetos ausencia del array**

**}**

**public String visualizar\_alumno(){**

**return "<AlumnoPerfeccionamiento>" + this.toString();**

**}**

**}**

**Resumiendo, en esta clase AlumnoPerfeccionamiento, implemente los siguientes métodos:**

**public void anyadir\_ausencia(String fecha, int horas)**

**public void mostrar\_ausencias\_alumno()**

**package alumnos;**

**public class SargentoTEL extends AlumnoPerfeccionamiento {**

**//atributo**

**private String destino;**

**//Constructor**

**public SargentoTEL (String nombre, String apellidos, String domicilio,**

**int orden, String destino){**

**super(nombre,apellidos,"2º CURSO CG-EB-MOP/TEL");**

**asignaturas = new Asignatura[7];**

**rellena\_asignaturas();**

**setEscalafon(orden);**

**this.destino=destino;**

**}**

**//metodos set/get destino**

**public void setDestino(String destiny){**

**destino=destiny;**

**}**

**public String getDestino(){**

**return destino;**

**}**

**//método propio**

**public double coeficiente(){**

**double coef=0;**

**//método que va restando 0,3 por cada hora de ausencia**

**return coef;**

**}**

**@Override**

**//método sobreescrito**

**public String visualizar\_alumno(){**

**String padre = super.visualizar\_alumno();**

**return padre + "|TEL|" + this.toString();**

**}**

**//implementacion métodos abstractos interfaces**

**public double calificacion\_parcial(){**

**double cp=0.0;**

**//método que hace la media de las notas de todas las asignaturas**

**return cp;**

**}**

**public double calificacion\_final(){**

**double nfinal=0-0;**

**//A la calificacion\_parcial le restamos el coeficiente**

**//método que devuelva la suma estos dos porcentajes**

**return nfinal;**

**}**

**}**

**Resumiendo, en esta clase SargentoTEL, implemente los siguientes métodos:**

**public double coeficiente()**

**public double calificacion\_parcial()**

**public double calificacion\_parcial()**

**package alumnos;**

**public class SargentoCAC extends AlumnoFormacion {**

**//atributo**

**private String provincia;**

**//Constructor**

**public SargentoCAC (String nombre, String apellidos, int orden, String Provincia){**

**super(nombre,apellidos,"24ª PROMOCION CG-EB-CAC/CIS");**

**asignaturas = new Asignatura[9];**

**rellena\_asignaturas();**

**setEscalafon(orden);**

**provincia=Provincia;**

**}**

**//metodos set/get provincia**

**public void setProvincia(String provincia){**

**this.provincia=provincia;**

**}**

**public String getProvincia(){**

**return provincia;**

**}**

**@Override**

**//método sobreescrito**

**public String visualizar\_alumno(){**

**String padre = super.visualizar\_alumno();**

**return padre + "|CAC|" + this.toString();**

**}**

**//implementacion métodos abstractos interfaces**

**public double calificacion\_parcial(){**

**double cp=0.0;**

**//método que hace la media de las notas de todas las asignaturas**

**return cp;**

**}**

**public double ipa(){**

**double informe\_alumno=10;**

**//metodo que vaya restando 1 por cada sancion del alumno;**

**//usando el método getAcumuladas de cada alumno**

**return informe\_alumno;**

**}**

**public double calificacion\_final(){**

**double nfinal=0.0;**

**//Calcular el 70% de la calificacion\_parcial**

**//Calcular el 30% del ipa**

**//método que devuelva la suma estos dos porcentajes**

**return nfinal;**

**}**

**}**

**Resumiendo, en esta clase SargentoCAC, implemente los siguientes métodos:**

**public** **double** calificacion\_parcial()

**public** **double** ipa()

**public** **double** calificacion\_final()

**package alumnos;**

**import java.util.Scanner;**

**import java.util.ArrayList;**

**public class Principal {**

**public static void main(String[] args) {**

**Scanner entrada = new Scanner(System.*in*);**

**ArrayList<AlumnoFormacion> af = new ArrayList<AlumnoFormacion>();**

**ArrayList<AlumnoPerfeccionamiento> ap = new ArrayList<AlumnoPerfeccionamiento>();**

**int opcion=0;**

**do{**

**opcion = Extras.*menu*();**

**switch(opcion){**

**case 1:{**

**//Añadir un objeto SargentoCAC al ArrayList af**

**/\* para construir el objeto SargentoCAC hay que pedir por teclado**

**\* nombre, apellidos, escalafón y provincia para el constructor**

**\* pediremos el numero de dni con una variable int y**

**\* usaremos el metodo setDni(int) heredado.**

**\* Pediremos la fecha de nacimiento con un string, comprobaremos**

**\* que es correcta con el método setFechaNacimiento(String)**

**\* Una vez que la fecha sea correcta, ejecutamos el método edad()**

**\*/**

**// independientemente de como fueron ingresados, el nombre se guardará**

**// en minusculas, y los apellidos en mayúsculas**

**break;}**

**case 2:{**

**//Añadir un objeto SargentoTEC al ArrayList ap**

**/\*para construir el objeto SargentoCAC hay que pedir por teclado**

**\* nombre, apellidos, escalafón y destino para el constructor**

**\* pediremos el numero de dni con una variable int y**

**\* usaremos el metodo setDni(int) heredado.**

**\* Pediremos la fecha de nacimiento con un string, comprobaremos**

**\* que es correcta con el método setFechaNacimiento(String)**

**\* Añadiremos un domicilio con el método setDomicilio(String)**

**\* Una vez que la fecha sea correcta, ejecutamos el método edad()**

**\*/**

**// independientemente de como fueron ingresados, el nombre se guardará**

**// en minusculas, y los apellidos en mayúsculas**

**break;}**

**case 3:{//Borrar objeto SargentoCAC del ArrayList af**

**/\* para borrar el objeto, preguntaremos por un patron de texto**

**\* que será de tamaño máximo el que tenga los apellidos**

**\* Recorreremos el ArrayList af, extrayendo el objeto con el método**

**\* get(int index),**

**\* y buscando este patrón en los apellidos. Cuando lo encontremos, pediremos**

**\* confirmación antes de eleminarlo mostrando sus datos personales.**

**\* Usaremos el método remove(int index)**

**\*/**

**break;}**

**case 4:{//Borrar objeto SargentoTEL del ArrayList ap**

**/\* para borrar el objeto, preguntaremos por un patron de texto**

**\* que será de tamaño máximo el que tenga los apellidos**

**\* Recorreremos el ArrayList ap, extrayendo el objeto con el método**

**\* get(int index),**

**\* y buscando este patrón en los apellidos. Cuando lo encontremos, pediremos**

**\* confirmación antes de eleminarlo mostrando sus datos personales.**

**\* Usaremos el método remove(int index)**

**\*/**

**break;}**

**case 5:{//Sancionar alumno**

**/\* Para sancionar un alumnos, primero preguntaremos por un número de escalafón**

**\* Recorreremos el ArrayList af, extrayendo, con el método get(int index)**

**\* el objeto y buscando este número de escalafón. Cuando lo encontremos**

**\* pediremos confirmacion y usaremos el método anyadiar\_sancion(String,int)**

**\*/**

**break;}**

**case 6:{//Anotar ausencia**

**/\* Para anotar una ausencia, primero preguntaremos por un número de escalafon**

**\* Recorreremos el ArrayList af, extrayendo, con el método get(int index)**

**\* el objeto y buscando este número de escalafón. Cuando lo encontremos**

**\* pediremos confirmacion y usaremos el método anyadiar\_ausencia(String,int)**

**\*/**

**break;}**

**case 7:{//Consulta sanciones alumnos**

**/\* Con un foreach recorreremos todo el arraylist af comprobando si los alumnos**

**\* tienen sanciones anotadas (método getCantidad). Si tiene sanciones**

**\* ejecutaremos el método mostrar\_sanciones\_alumno(),**

**\*/**

**break;}**

**case 8:{//Consulta ausencias alumnos**

**/\* Con un foreach recorreremos todo el arraylist ap comprobando si los alumnos**

**\* tienen ausencias anotadas (método getCantidad). Si tiene ausencias**

**\* ejecutaremos el método mostrar\_ausencias\_alumno(),**

**\*/**

**break;}**

**case 9:{//Listado calificaciones finales todos los alumnos**

**ArrayList<Alumno> a = new ArrayList<Alumno>();**

**/\* Copiaremos los dos arraylist af y ap en el arraylist<Alumno> con**

**\* sendos bucles for normales.**

**\* primero los alumnos de formación y seguido los de perfeccionamiento**

**\* Una vez copiados, recorreremos al ArrayList<Alumno> con un foreach para listarlos**

**\* En el listado usaremos el método calificacion\_final, mostrando antes de**

**\* esta calificacion\_final los siguiente datos personales: apellidos,**

**\* nombre, dni, edad, curso, escalafón**

**\* Como extra, mostrará tambien el destino o provincia según corresponda**

**\* [OPERADOR INSTANCEOF] + conversión explícita (hija)padre**

**\* despues de visualizar estos datos personales y la calificación final de cada**

**\* alumno, ejecutará tambien el método visualizar\_alumnos que se ha ido**

**\* sobreescribiendo, añadiendole (anteponiendo) sus iniciales a la**

**\* cadena que devuelve**

**\*/**

**break;}**

**default:{**

**System.*out*.println("\nXXX OPCION INCORRECTA XXX");**

**break;}**

**}**

**}while (opcion!=0);**

**}**

**}**

**Resumen, desarrollar los 9 case:**

**EXTRA PARA NOTA: Añada tres bloques (1 de cada) try-catch, para capturar y tratar las posibles excepciones siguientes en algún fragmento de código que pueda lanzarlas**

1. **ImputMismatch**
2. **IndexOutOfBounds**
3. **NullPointer**