

Nombre:	Nota
Curso:	

Cosas permitidas y no permitidas

- El código entregado tiene que compilar sin problemas
- Cada apartado puntuable se calificará de 0-10. Si hubiese algún error muy grave, directamente podrá calificarse de 0. Si el error es inasumiblemente grave (se nota que no se tiene ni idea de lo que se está haciendo), la nota máxima del examen será de 4.
- En algunos puntos, se distingue el peso relativo de cada sección dentro de la propia pregunta
- No puedes usar Internet ni list.sort, ni las funciones agregadas de las listas, incurrir en esto repercutirá en una nota de 0 en todo el examen



Tiempo estimado

Dificultad

Tipo de actividad

2 horas

Baja

Evaluación
Nota máxima: 10



Se desea implementar una aplicación para manejar las notas de los estudiantes. Para ello se decide realizar las siguientes clases:

- (3 puntos) **Classroom**: Tiene una lista de estudiantes
- (2 puntos) **Student**: Es un estudiante.
- (2 puntos) **Notes**: Contiene las notas de las asignaturas tales como matemáticas, ciencia, historia y lengua.
- (3 puntos) **Statistics**: Es una clase de la que no se va a crear ningún objeto, sirve para calcular estadísticas

Los atributos de las clases, que son todos privados, son las siguientes:

Classroom

- Una lista de estudiantes
- Un nombre

(Peso: 3) Student

- Nombre
- Edad
- Género (enum)
- Altura
- Peso
- Una instancia de la clase Notes

Notes

- Contiene las notas de matemáticas, ciencia, historia y lengua.
- Las notas pueden tener decimales
- Se tendrá en cuenta positivamente que se pueda manejar muchas más notas sin variar los algoritmos.

Statistics

- No contiene nada

Los métodos de las clases, que son todos públicos, son las siguientes. Como podrás observar, hay cosas que se establecen, y otras que se dejan abiertas:

Classroom

- (Peso: 1) Constructor vacío
- (Peso: 3) Constructor al que se le pasa una lista de estudiantes, y almacenará una copia de esa lista
- (Peso: 4) Los setters y los getters que consideres necesario
- (Peso: 2) AddStudent(Student)
- (Peso: 2) RemoveStudentAt(int index)
- (Peso: 2) ContainsStudentWithName(string)
- (Peso: 4) RemoveStudentsWithName(string)

Student

- (Peso: 1) Constructor vacío
- (Peso: 2) Constructor al que le paso nombre y edad
- (Peso: 4) Los setters y los getters que consideres necesario
- (Peso: 2) GetIMC() Esto se calcula como $IMC = peso / (altura^2)$
- (Peso: 2) Obtener la media de todas sus asignaturas
- (Peso: 1) Es Mayor de Edad?



- (Peso: 4) **Clone**. Ojo, ..., esta función clona al alumno entero (**copia en profundidad**), ..., cuidado con las notas. Es decir, si clono un alumno, y modifico las notas del alumno clonado, las notas del original no se deben ver afectadas.

Notes

- (Peso: 1) Constructor vacío
- (Peso: 2) Los setters y los getters que consideres necesario
- (Peso: 6) **GetQualificationForSignature**(algo que indique de qué asignatura quiero saber la nota, ..., y no valen 4 booleanos)
- (Peso: 6) **SetQualificationForSignature**(?????)
- (Peso: 2) Obtener la media de todas las asignaturas
- (Peso: 4) Obtener la asignatura con la nota más alta
- (Peso: 1) Obtener la asignatura con la nota más baja
- (Peso: 4) Obtener la calificación más baja
- (Peso: 1) Obtener la calificación más alta

Statistics

- (Peso: 1) **GetAverageIMC(Classroom)** → Devuelve el IMC medio de todo el alumnado.
- (Peso: 1) **GetBestStudent(Classroom)** → Devuelve el alumno que tenga la mayor nota en alguna de las asignaturas.
- (Peso: 2) **GetYoungestStudent(Classroom)** → Devuelve el alumno más joven
- (Peso: 3) **GetSortedStudentsForSignature(Classroom, asignatura??)** → Devuelve una lista de estudiantes ordenada de mayor a menor teniendo en cuenta su calificación en la asignatura especificada.
- (Peso: 3) **GetStudentsWithGender(...)** → Devuelve una lista de estudiantes con el género indicado ordenados por edad
- (Peso: 6) **GetStatistics(Classroom)** → “A saber cómo implementas esto”, ... No te preocupes por detalles insignificantes como “<” o “<=”. Esta función tiene que devolver en la misma llamada:
 - el número de estudiantes que ha sacado como nota media más de un 9
 - el número de estudiantes que ha sacado como nota media entre 7 y 9
 - el número de estudiantes que ha sacado como nota media entre 5 y 7
 - el número de estudiantes que ha sacado como nota media entre 3 y 5
 - el número de estudiantes que ha sacado como nota media entre 0 y 3
 - No sé lo que será lo que devuelva, pero el que llame a esta función, sea lo que sea lo que le devuelva, no puede dar lugar a dudas

Otras consideraciones:

- Desde la clase “Classroom” debemos de poder acceder a todos los/as alumnos/as.
- Cuidado con las funciones tipo “Devuelve el alumno más joven”. Esto debería devolver al objeto alumno, no su nombre.
- Las funciones de Statistics se deben implementar en esa clase, no en Classroom, es decir, Statistics no puede ser un bypass a Classroom
- Ninguna de las funciones de Statistics devuelve un string
- El nombre de las asignaturas no es un string