

UNIVERSIDAD TECNOLÓGICA DEL PERÚ

TALLER DE PROGRAMACIÓN - PRÁCTICA CALIFICADA 03 (04/07/2024)

Apellidos y Nombres:

Pregunta 1: implemente en Java el siguiente diagrama de clases:

CompresorCadena
+comprimir(cadena: String): String +descomprimir(cadena: String): String

comprimir() comprime una cadena anexando a cada caracter el número de repeticiones (si son más de 2). Por ejemplo, la cadena "jaavvvaaaaa" se comprime como "jaav3a5"
descomprimir() descomprime una cadena repitiendo los caracteres en base a su conteo. Por ejemplo, la cadena "go7l3" se descomprime como "gooooooooooIII"
Asuma que las cadenas solo contienen caracteres alfabéticos y que una letra no tiene más de 9 repeticiones.

Pregunta 2: implemente en Java el siguiente diagrama de clases:

Velocista
-dni: String -nombre: String -marcas: double[5] +calcularPromedio(): double +haClasificado(): boolean +toString(): String

Crear un constructor completo y getters & setters para todos los atributos.

calcularPromedio() retorna el promedio de las cuatro mejores marcas del velocista.
haClasificado() indica si el velocista ha alcanzado una marca promedio máxima de 12.5 segundos.
toString() debe devolver: "dni - nombre [marcas]"

Pregunta 3: implemente en Java el siguiente diagrama de clases:

Competencia
-velocistas: Velocista[10] -contador: int +agregarVelocista(velocista: Velocista): void +reportarTablaPosiciones(): String

Crear un constructor vacío e inicializar los atributos. Crear getters & setters para todos los atributos.

Se puede registrar un máximo de 10 velocistas.
reportarTablaPosiciones() debe devolver una cadena conteniendo un reporte ASCII con la tabla de posiciones y la indicación de quiénes clasificaron.

UNIVERSIDAD TECNOLÓGICA DEL PERÚ

TALLER DE PROGRAMACIÓN - PRÁCTICA CALIFICADA 03 (04/07/2024)

Apellidos y Nombres:

Pregunta 1: implemente en Java el siguiente diagrama de clases:

CompresorCadena
+comprimir(cadena: String): String +descomprimir(cadena: String): String

comprimir() comprime una cadena anexando a cada caracter el número de repeticiones (si son más de 2). Por ejemplo, la cadena "jaavvvaaaaa" se comprime como "jaav3a5"
descomprimir() descomprime una cadena repitiendo los caracteres en base a su conteo. Por ejemplo, la cadena "go7l3" se descomprime como "gooooooooooIII"
Asuma que las cadenas solo contienen caracteres alfabéticos y que una letra no tiene más de 9 repeticiones.

Pregunta 2: implemente en Java el siguiente diagrama de clases:

Velocista
-dni: String -nombre: String -marcas: double[5] +calcularPromedio(): double +haClasificado(): boolean +toString(): String

Crear un constructor completo y getters & setters para todos los atributos.

calcularPromedio() retorna el promedio de las cuatro mejores marcas del velocista.
haClasificado() indica si el velocista ha alcanzado una marca promedio máxima de 12.5 segundos.
toString() debe devolver: "dni - nombre [marcas]"

Pregunta 3: implemente en Java el siguiente diagrama de clases:

Competencia
-velocistas: Velocista[10] -contador: int +agregarVelocista(velocista: Velocista): void +reportarTablaPosiciones(): String

Crear un constructor vacío e inicializar los atributos. Crear getters & setters para todos los atributos.

Se puede registrar un máximo de 10 velocistas.
reportarTablaPosiciones() debe devolver una cadena conteniendo un reporte ASCII con la tabla de posiciones y la indicación de quiénes clasificaron.