

Ejercicio 5

La funcionalidad total del programa debe ser que se cree un sistema simulador con edificios, niveles, prioridades y operadores para atender las llamadas de un callcenter. En cada iteración del simulador se genera una nueva llamada aleatoriamente para ser asignada aleatoriamente a un operador. El operador solo puede tener una llamada actual y una llamada espera. Solo toma la llamada en espera, si tiene 3 minutos restantes en la llamada actual, sino la rechaza y la pasa a otro operador aleatoriamente. Se puede decidir si generar una nueva iteración o ver los datos del call center en su totalidad.

Clase drivercallcenter:

- **Objetivo:** La función es ser el driver de nuestro programa. Aquí se interactuara con el usuario y se pedirá la información necesaria para ejecutar las funcionalidades del programa.
- **Propiedades:**
 - Scanner llamado leer para recibir datos
 - Int edificios dato que se recibirá para ser enviado como parámetro luego.
 - Int niveles dato que se recibirá para ser enviado como parámetro luego.
 - Int operadores dato que se recibirá para ser enviado como parámetro luego.
 - Sistemasmulador callcenter será la instancia de nuestro sistema en el Main.
 - String respuesta, será la respuesta al menu para ejecutar una funcionalidad según sea el numero.
- **Metodos:**
 - public static void main(String args[]) Para ejecutar las funcionalidades de las demas clases y de nuestro programa.

Clase Scanner:

- **Objetivo:** Ser una instancia para obtener datos del usuario.
- No tiene propiedades.
- **Métodos:**
 - Scanner() para instanciar el scanner.
 - nextInt() lee el siguiente int ingresado.
 - Next() lee el siguiente string ingresado.

Clase sistemasimulador:

- **Objetivo:** La principal función de esta clase es poder inicializar la cantidad de edificios, cantidad de niveles por edificio y cantidad de operadores por nivel. La cantidad de operadores estará dividida en tres filas de distintas prioridades. Siendo 1 baja, 2 media, y 3 alta.
- **Propiedades (Todas son privadas.):**
 - int[][][] sistema; Aquí se almacenan los datos en un arreglo int, el primer arreglo es la cantidad de edificios, el segundo la cantidad de niveles que debe ser por lo menos 3, el número de filas que son 3 y por último, la cantidad de operadores en dicha fila.

- Int edificios; La cantidad de edificios en el sistema.
- Int niveles; La cantidad de niveles en el sistema.
- Int operadores; La cantidad de operadores en el sistema.
- Int filas; La cantidad de filas en el sistema.
- Int minutos; La cantidad de minutos en el simulador. El simulador inicia en 0 minutos.
- Métodos:
 - Randint(min: int, max: int): Recibe 2 valores un mínimo y un máximo para devolver un valor aleatorio en un intervalo cerrado entre ambos. Devuelve un valor int.
 - Sistemasimulador(int ed, int niv, int oper): Recibe los datos para instanciar un nuevo sistema simulador con la cantidad de edificios, niveles y operadores para generar un sistema.
 - Mostraroperadores(): Devuelve un valor string que muestra los estados de los operadores y las estadísticas solicitadas por el programa.
 - Simularminuto(): Devuelve un valor String de la simulación de un minuto, luego de ejecutar la simulación sobre generar una llamada aleatoria, verificar a los agentes y actualizar las llamadas.

Clase operador:

- Objetivo: La principal función de esta clase es poder crear un objeto de tipo operador que sea un trabajador de la empresa con las propiedades de su ubicación para poder tomar llamadas que se generaran aleatoriamente por el sistema simulador.
- Propiedades (Todas son privadas.):
 - Int edificio; El edificio en el que se encuentra.
 - Int nivel; El nivel en el que se encuentra.
 - Int prioridad; La prioridad en el que se encuentra.
 - Int numero; El número de posición en el que se encuentra.
 - String identificación; El código de identificación del empleado según donde se encuentra.
 - Int llamadaactual; La cantidad de minutos en la llamada actual. Si es igual a 0, entonces se encuentra disponible.
 - Int llamadaespera; La cantidad de minutos que hay en la llamada de espera. Si es igual a 0, entonces no tiene llamada en espera.
 - Int minutostotales; La cantidad de minutos totales en los que el operador ha atendido una llamada.
 - Int llamadasterminadas; La cantidad de llamadas totales atendidas por el operador.
 - Int static llamadastotales; La cantidad de llamadas totales atendidas por todos los operadores.
 - Int static totaloperadores; La cantidad total de operadores en el sistema.
- Métodos:
 - operador(int ed, int niv, int oper): Recibe los datos para instanciar un nuevo operador según su ubicación.

- Getllamadaactual(); Devuelve un valor de tipo int que es el tiempo restante en la llamada actual.
- Setllaamadaactual(int actual); Este método recibe el valor de una nueva llamada para ser asignada como la actual y devuelve un String mostrando el mensaje que llamada ha sido tomada.
- Getllamadaespera(); Devuelve un valor de tipo int que es el tiempo restante en la llamada de espera.
- Setllaamadaespera (int espera); Este método recibe el valor de una nueva llamada para ser asignada a la lista de espera del operador.
- Getidentificacion(); Este método devuelve la identificación del operador para ser comparado en el sistema simulador con el valor aleatorio calculado.
- Info(); Este método devuelve un string que muestra los datos de estatus y estadísticas del operador.
- Operadorminuto(); Este método no devuelve ningún valor, simplemente resta un minuto a la llamada actual si es que se encuentra en una y si tenía una llamada en espera y ahora esta libre, se le asigna automáticamente.

Clase DecimalFormat:

- El objetivo de esta clase es crear un formato que muestre únicamente 2 dígitos decimales.
- Sus métodos serán DecimalFormat para instanciar un tipo de formato y format() para convertir un número a un string de dicho formato y que lo devuelva.