

PSP - Examen 1a Evaluación

Gestor de procesos

Proyecto base

Las clases proporcionadas y los archivos de prueba no se pueden modificar.

En la entrega del ejercicio no se puede usar el método `inheritIO()`.

psp.examenes.Examen1_GestorProcesos

Disponemos de dos clases auxiliares en el paquete psp.examenes, ya implementadas:

- Examen1_ContadorLineas
- Examen1_GeneradorPalabras

Tu tarea es programar la clase principal `Examen1_GestorProcesos`, que realizará las siguientes operaciones utilizando procesos:

Parte A — Lanzar Examen1_ContadorLineas

Lanza un proceso que ejecute la clase `Examen1_ContadorLineas`, pasándole como **argumento** el archivo `datos/textoEntrada.txt`.

La salida estándar del proceso debe recogerse y mostrarse por consola precedida del prefijo:

```
conteo>
```

La salida de error debe **redirigirse** al archivo `datos/erroresContador.txt`, sobrescribiendo el contenido anterior. Cuando el proceso finalice, debes obtener y mostrar su código de salida.

Parte B — Lanzar Examen1_GeneradorPalabras

Lanza un proceso que ejecute la clase `Examen1_GeneradorPalabras`.

Su entrada estándar debe provenir del archivo `datos/listaPalabras.txt`.

Sus salidas estándar y de error deben recogerse y mostrarse por consola, ambas precedidas por:

```
generador>
```

Espera a que termine y muestra su código de finalización.

Fábrica de bicicletas

Crea una aplicación para simular el funcionamiento de una fábrica de bicicletas.

En la fábrica de bicicletas BalmisBikes tenemos tres tipos de trabajadores que deben coordinarse para cumplir con las siguientes tareas:

- Operarios de cuadros: Producen cuadros de bicicleta. Hay **3 operarios** de cuadros.
- Operarios de ruedas: Producen pares de ruedas. Hay **2 operarios** de ruedas.
- Montadores: Montan bicicletas usando 1 cuadro y 2 ruedas (un par). Hay **4 montadores**.

La producción sigue este proceso:

- Los operarios pueden producir un cuadro o un par de ruedas y dejarlos en su almacén, pero si el almacén está lleno, deben esperar.
 - El almacén de cuadros tiene una capacidad de **4 unidades** y el de pares de ruedas tiene una capacidad de **10 unidades (5 pares)**.
 - Los operarios de cuadros tardan entre **2h y 4h** en producir un cuadro
 - Los operarios de ruedas tardan entre **1h y 3h** en producir un par de ruedas.
- Los montadores solo pueden armar una bicicleta si hay, al menos, 1 cuadro y 1 par de ruedas disponibles. Si no es posible, deben esperar.
 - Los montadores tardan entre **media hora y 2h** en montar una bicicleta.

Se pide realizar una aplicación en Java, siguiendo el modelo productor-consumidor, que simule el funcionamiento de la fábrica, mostrando por consola los mensajes necesarios para entender qué está pasando en cada momento.

La simulación debe ejecutar un mínimo de **20 montajes** exitosos de bicicletas antes de finalizar.

Información del proceso

La aplicación debe mostrar claramente qué está pasando en cada momento.

Los mensajes deben indicar el estado: producción, espera, montaje y retirada de piezas, con el nombre del hilo responsable.

La salida por consola debe permitir distinguir fácilmente qué trabajador produce qué y cuándo (colores opcionales, nombres obligatorios).

El nombre de los trabajadores se puede establecer como "OperarioCuadros-1", "OperarioRuedas-2", "Montador-3", etc.

Simulación en escala

Se pide simular en escala el proceso de cada trabajador.

Cuando se indica que un trabajador tarda entre X e Y horas en realizar su tarea, puede tardar un tiempo aleatorio (todo el rango posible de tiempo) entre esos valores, no solo horas enteras.

Cada hora real = 100 ms en la simulación.

⚠ Gestión de excepciones en el código

Controla y maneja excepciones en tu código (no las escales), mostrando la pila cuando sea necesario.

⚡ Parametrización del código

Ten en cuenta que el número de operarios, capacidades de almacén, tiempos de producción y montaje, etc., deben estar parametrizados en constantes al inicio del código para facilitar su modificación.

Es recomendable probar diferentes configuraciones para comprobar el correcto funcionamiento de la aplicación en distintos escenarios.