

Práctica 6. Uso básico de excepciones

Objetivos

- Practicar el uso básico de bloques `try-catch`.
- Practicar la creación y el uso de excepciones creadas por el programador.

Desarrollo

Modificar la práctica 5 en dos aspectos:

- Coger las excepciones lanzadas por métodos del paquete `fundamentos`
- Modificar las clases implementadas por el alumno para que lancen excepciones bajo alguna condición de error.

Documento de requisitos

Excepciones lanzadas por el paquete `fundamentos`

Los métodos `Lectura.leeDouble` y `Lectura.leeInt` lanzan la excepción `NumberFormatException` cuando el valor escrito en la entrada correspondiente no es un número (real o entero respectivamente).

Se desea modificar el programa principal de la práctica 5 para que coja la citada excepción de forma que se escriba un mensaje de error para el usuario. Después de mostrar el citado error, el programa no deberá finalizar, sino que deberá volver a mostrar el menú principal.

Utilización de excepciones propia en la clase `RegistroTrabajos`

Se desea modificar el comportamiento del método `RegistroTrabajos.añade()` (el equivalente a este método dependiendo de los nombres de la clase y el método elegidos por el alumno) cuando se de la circunstancia de que no sea posible registrar más trabajos debido a que se ha alcanzado el tamaño máximo del array.

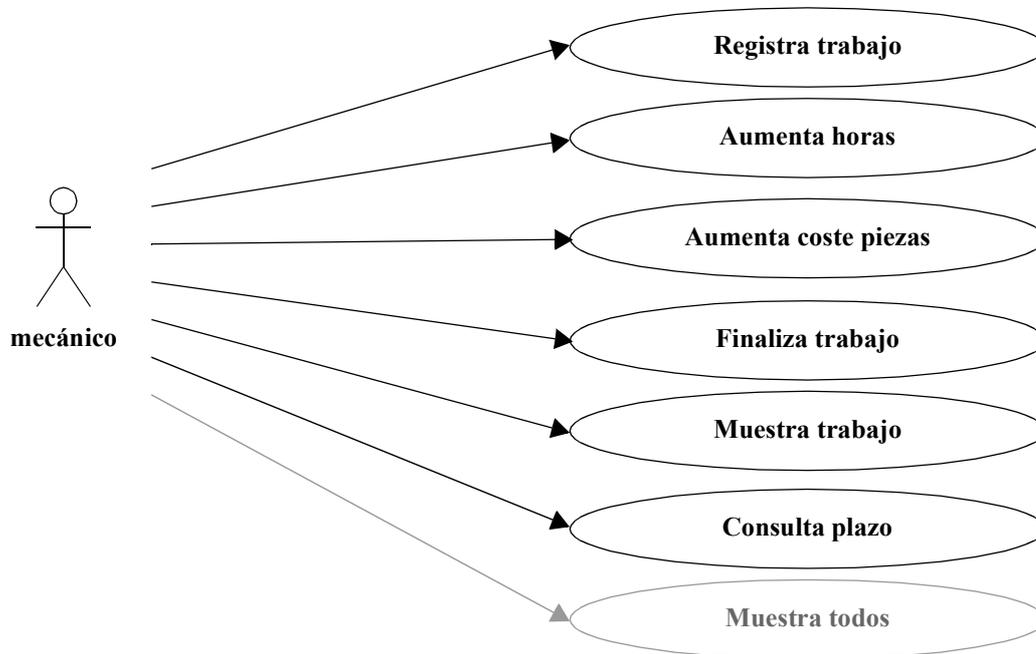
En la implementación realizada en la práctica 5, este método retornaba un booleano para indicar si se había registrado con éxito el trabajo. En la modificación propuesta el método deberá lanzar la excepción `RegistroLleno` cuando no sea posible registrar más trabajos.

La excepción `RegistroLleno` deberá ser creada por el alumno dentro de la misma clase `RegistroTrabajos`.

La excepción `RegistroLleno` no deberá provocar la terminación de la aplicación, sino que deberá mostrarse un mensaje de error y retornar al menú principal.

Casos de uso

Las operaciones que se desea que los mecánicos puedan realizar son las descritas en los siguientes casos de uso (los casos en color más oscuro son los que sufren alguna modificación respecto a la práctica 5):



Caso de uso "Registra trabajo":

1. El mecánico selecciona la opción "Registra trabajo".
2. La aplicación muestra una ventana que permite introducir el tipo de trabajo y su descripción.
3. El mecánico introduce los datos.
4. La aplicación añade el trabajo a la lista de trabajos registrados y muestra el identificador asignado al trabajo.
 - en el caso de que no sea posible registrar el trabajo por haberse alcanzado el número máximo de trabajos, se notifica el error y finaliza el caso de usos

Caso de uso "Aumenta horas":

1. El mecánico selecciona la opción "Aumenta horas".
2. La aplicación muestra una ventana que permite introducir el identificador del trabajo y el número de horas.
3. El mecánico introduce los datos.
4. En el caso de existir un error de formato en el identificador o en el número de horas, se muestra un mensaje de error y finaliza el caso de uso
5. La aplicación aumenta las horas al trabajo.
 - Se notifica un error si el trabajo ya está finalizado, el número de horas es negativo o el identificador es incorrecto.

Caso de uso "Aumenta coste piezas":

1. El mecánico selecciona la opción "Aumenta coste piezas".
2. La aplicación muestra una ventana que permite introducir el identificador del trabajo y el coste de las piezas.
3. El mecánico introduce los datos.

4. En el caso de existir un error de formato en el identificador o en el coste de las piezas, se muestra un mensaje de error y finaliza el caso de uso
5. La aplicación aumenta el coste de las piezas al trabajo.
 - Se notifica un error si el trabajo no es una reparación, o ya está finalizado, o el coste es negativo o el identificador es incorrecto.

Caso de uso "Finaliza trabajo":

1. El mecánico selecciona la opción "Finaliza trabajo".
2. La aplicación muestra una ventana que permite introducir el identificador del trabajo.
3. El mecánico introduce el identificador.
4. En el caso de existir un error de formato en el identificador, se muestra un mensaje de error y finaliza el caso de uso
5. La aplicación registra el trabajo como finalizado.
 - Se notifica un error si el identificador es incorrecto.

Caso de uso "Muestra trabajo":

1. El mecánico selecciona la opción "Muestra trabajo".
2. La aplicación muestra una ventana que permite introducir el identificador del trabajo.
3. El mecánico introduce el identificador.
4. En el caso de existir un error de formato en el identificador, se muestra un mensaje de error y finaliza el caso de uso
5. La aplicación muestra el identificador, la descripción y el precio del trabajo.
 - Se notifica un error si el identificador es incorrecto.

Caso de uso "Consulta plazo":

1. El mecánico selecciona la opción "Consulta plazo".
2. La aplicación muestra una ventana que permite introducir el identificador del trabajo.
3. El mecánico introduce el identificador.
4. En el caso de existir un error de formato en el identificador, se muestra un mensaje de error y finaliza el caso de uso
5. La aplicación muestra el plazo máximo en el que debería estar finalizado el trabajo.
 - Se notifica un error si el identificador es incorrecto.

Caso de uso "Muestra todos":

1. El mecánico selecciona la opción "Muestra todos".
2. La aplicación muestra una ventana que muestra los datos de todos los trabajos: para todos los trabajos se muestra: identificador, horas, precio y plazo. En el caso de las reparaciones se muestra además el dinero gastado en material.

Entregar

- Código desarrollado (todo el código, no únicamente las clases modificadas)