Algoritmos y Programación I
2024-2 - Evaluación Práctica 1 - Grupo 1

Unidad 1: Introducción a la Programación y al lenguaje de programación

OE1.1. Analizar problemas mediante la especificación, a través de contratos, de entradas, salidas, ejemplos y casos de prueba.
OE1.2. Modelar información relevante a la solución del problema empleando variables, constantes, tipos de datos primitivos y cadenas de texto.
OE1.3. Resolver problemas utilizando estructuras de control: instrucciones secuenciales, subrutinas, condicionales y repetitivas.
OE1.5. Utilizar un ambiente de desarrollo (incluyendo la compilación y ejecución de programas desde consola) y un espacio de trabajo predefinido, para construir la solución de un problema.
OE1.4. Utilizar operadores (de asignación, aritméticos, relacionales, de cadenas y lógicos), estructuras contenedoras lineales de tamaño fijo (de tipos de datos primitivos) y cadenas de texto en la construcción de soluciones.
OE1.6. Codificar en lenguaje Java la solución a un problema a partir de los contratos de solución propuestos en las etapas de análisis y diseño.
OE1.7. Utilizar objetos e invocar métodos estáticos de clases del API de Java en la construcción de soluciones implementadas con interfaces gráficas por consola.
OE1.8. Interpretar y resolver errores producidos en tiempo de ejecución (ej.: posición por fuera del rango en una estructura contenedora, llamados u operaciones con objetos que no han sido construidos, etc.).

Problema: BurgerTown

BurgerTown, un nuevo restaurante de comida rápida, es un emprendimiento de Carolina, una egresada de Administración de Empresas, de la Universidad Icesi. Carolina decidió abrir su propio negocio tras finalizar la universidad. Carolina sabe que estás tomando un curso de Algoritmos y Programación I, por lo que te ha pedido ayuda para desarrollar un sistema que le permita registrar las ventas diarias en su restaurante.

Requisitos:

Al iniciar el programa, se debe solicitar al usuario el número de platos diferentes vendidos durante un día. (Funcionalidad proporcionada en el código base)

1. El programa debe **solicitar el precio ($)** y la **cantidad** de cada uno de los platos vendidos y almacenará estos datos respectivamente en arreglos.
2. El programa debe contar con una funcionalidad que permita calcular la **cantidad total de platos vendidos durante el día**.
3. El programa debe contar con una funcionalidad que permita calcular el **precio promedio de los platos vendidos durante el día**.
4. El programa debe contar con una funcionalidad que permita calcular las **ventas totales (dinero recaudado) durante el día**.
5. El programa debe contar con una funcionalidad que permita consultar el número de platos que **hayan superado un límite mínimo de ventas proporcionado por el usuario**. Por ejemplo: El usuario ingresa $300.000 y el programa retorna que solo 3 platos (ejemplo) alcanzaron ese mínimo de ventas.

Especificaciones:

- El programa se debe desarrollar usando la estructura de paquetes (carpetas) vista en clase.
- El programa debe contar con métodos para cada funcionalidad.
- Su profesor le ha proporcionado un código base con la estructura del menú principal y los encabezados de los métodos a desarrollar. Puedes optar por realizar cambios según lo consideres necesario.

Documentación:

- El programa debe contar con **contratos** en los métodos asociados a al menos 3 de las funcionalidades detalladas en los requisitos 1 al 5.
- Se debe generar la documentación del programa empleando el comando javadoc.

Entregables:

IMPORTANTE: Debe realizar su entrega a través de GitHub Classroom: GitHub
PLAZO: Desde las 11:00 AM del viernes 13 de septiembre 2024 hasta la 1:00 PM del viernes 13 de septiembre 2024

Tenga en cuenta que su proyecto debe presentar la estructura base presentada a continuación:

BurgerTown/
src/
bin/
doc/

Dentro de los directorios src/ y bin/ estarán presentes estos directorios, representando cada uno de sus paquetes:

ui/

Contenido en los directorios:
El directorio src (source code) contiene los archivos .java dentro del directorio ui (por ahora).
El directorio bin (binary files) contiene los archivos .class en el directorio ui.
El directorio doc contiene el análisis, diseño y documentación del problema: En esta entrega debe tener la subcarpeta API y el javadoc generado de su implementación.

Rubrica: Enlace

Nota	5	4		3	2	1
Uso del idioma inglés	El código, comentarios y diagramas tienen un uso del idioma ingles	El código, comentarios y diagramas estan en ingles, con algunos	El código, comentarios y diagramas estan en ingles, con errores.	El código, comentarios y diagramas estan en ingles, con	El código, comentarios y diagramas estan en	

	sin errores evidentes.	errores menores.		errores graves.	español	
Buenas prácticas	- El código está correctamente indentado - El nombre de la clase empieza en mayúscula - El nombre de la clase es adecuado para el problema - Los nombres de las variables son adecuados (incluye los parámetros) - Los nombres de las variables empiezan en minúscula - Los nombres de las subrutinas empiezan en minúscula					
Compilación y Ejecución del Código	El programa compila y se ejecuta sin errores.	El programa compila, pero hay unos errores en tiempo de ejecución		Al incluír unas pequeñas modificaciones el programa compila	-	El programa no compila y hay que hacer modificaciones mayores para hacerlo compilar

Documentación (20%)		**Codificación (80%)**							**Total**	**Bono**
El programa tiene definidos los contratos de al menos **3** de las subrutinas / métodos relacionados con los requerimientos 1 a 5	Se genera una versión actualizada del API del programa utilizando adecuadamente el comando javadoc	Buenas prácticas	Compilación y Ejecución del Código	El programa debe solicita el precio ($) y la cantidad de cada una de los platos vendidos y los almacena respectivamente en arreglos.	El programa permite calcular la cantidad total de platos vendidos durante el día.	El programa permite calcular el precio promedio de los platos vendidos durante el día.	El programa permite calcular las ventas totales (dinero recaudado) durante el día.	El programa permite consultar el número de platos que en el día hayan superado un límite mínimo de ventas proporcionado por el usuario		Uso del idioma inglés en la codificación (código del programa) y documentación (contratos)
15,0%	**5,0%**	**5,0%**	**5,0%**	**10,0%**	**15,0%**	**15,0%**	**15,0%**	**15,0%**	**100%**	**10%**