

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN,

CONTROL Y EVALUACIÓN DE RECURSOS INFORMÁTICOS

LICENCIATURA EN INGENIERÍA DE SISTEMAS DE INFORMACIÓN

FUNDAMENTOS DE PROGRAMACIÓN

Proyecto2020

Profesora: Felicita Castillo de KrolIntegrantes

Batista, Jeimy 8-970-2015

Espinosa ,Christian 20-24-5225

Chin ,Willie 8-973-698

Tejada ,Ángel 8-969-974

Eduardo Díaz 4-802-1774

Castillo ,Orlando 8-956-270

Solis, Michael 8-958-1219

ll SEMESTRE

**Definición o dominio del Problema:**

Una fábrica de paletas emplea 20 carritos especiales para la carga, venta y

distribución del producto. Estos carritos cargan una computadora. Todos los días

temprano específicamente a las 7:00 a.m. salen los carritos surtidos inicialmente de

100 paletas para cada uno de los sabores como: naranja, piña, limón, tamarindo y

coco, regresando a las 4:00 p.m. de la tarde. Inmediatamente se requiere conocer

cuántas paletas quedan de cada sabor para poder llenar los carritos con la cantidad

inicial de todos los días para salir temprano a la venta. Se desea que desarrolle un

programa para registrar las ventas y conocer al final del día:

• Cuánto queda en existencia de cada sabor de paleta para cada carrito.

• Cuánto se requiere de cada sabor para completar las 100 paletas en cada

carrito.

• Cuánto dinero se hizo en las ventas en cada carrito por cada sabor de paleta.

• Además, se necesita conocer el sabor que más se vende y que número de

carrito la vende. Contemple que pueden existir igualdad.

**Metodología de Trabajo POO**

**Análisis**

|  |  |
| --- | --- |
| Entrada de datos | private int fila = 20, col = 5; |
| Proceso | Métodos: setAcc(int carrito, int sabor, int a[][], int cantidad)  joke(int carrito, int sabor, float b[][], float precio, int cantidad)  [ ]posicionmayor(int a[ ][ ])  mayor(int a[ ][ ])  [ ]posicion(int maximo, int a[ ][ ])  [ ]posicion1(int maximo, int a[ ][ ])  PaletaEjemplo() |
| Salida | "\nSabores que quedan por cada carrito "  "Sabor 1 2 3 4 5"  ("\nCarrito " , (i+1) ,": ", (20-x [i][j]));  "\n\nPaletas en cada sabor que se requieren para completar las 100 paletas en cada carrito"  "Sabor 1 2 3 4 5"  "\nCarrito " , (i+1) + ": ", x [i][j];  "\n\nDinero ganado por cada carrito en cada sabor"  "Sabor 1 2 3 4 5 "  "\nCarrito " , (i+1) + ": ",z [i][j];  "\n\nEl sabor que más vende es el número: ",p[1]," y el carrito que más vende es el ",p[0],"."  "\nPosiciones maximas que encuentran carritos: ",sum  "\nPosiciones maximas que encuentran sabores: “,sum |

Identificar las abstracciones (clases)

Nombre de la clase: EjemploPaleta

**Diseño de clases (UML)**

Métodos

Datos o atributos

Nombre de la clase

|  |
| --- |
| EjemploPaleta |
| -private int fila = 20, col = 5; |
| public void setAcc (int carrito, int sabor, int a[][], int cantidad)  public void joke(int carrito, int sabor, float b[][], float precio, int cantidad)  public int []posicionmayor(int a[][])  public int mayor(int a[][])  public int []posicion(int maximo, int a[][])  public int []posicion1(int maximo, int a[][])  EjemploPaleta() // método constructor |

Dibujo del objeto:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |

obj

|  |
| --- |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

|  |
| --- |
| 0 |
| 1 |
| 2 |
| 3 |
| . |
| . |
| 19 |

x

EjemploPaleta()

public void setAcc (int carrito, int sabor, int a[][], int cantidad)

z

poss

public int []posicion(int maximo, int a[][])

public int mayor(int a[][])

public int []posicionmayor(int a[][])

public void joke(int carrito, int sabor, float b[][], float precio, int cantidad)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |

|  |
| --- |
| 0 |
| 1 |
| 2 |
| 3 |
| . |
| . |
| 19 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | . | . | . | 19 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |

mayorp

0

0

1

0

pos