using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using WarehouseLibrary;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

// Создаем экземпляр WarehouseManager

var warehouseManager = new WarehouseManager();

// Тест 1: Подсчет общего количества товара на всех складах

int productID = 1; // Пример ID товара

int totalQuantity = warehouseManager.CalculateTotalQuantityForProduct(productID);

Console.WriteLine($"Общее количество товара с ID {productID} на всех складах: {totalQuantity}");

// Тест 2: Подсчет количества товара на конкретном складе

int warehouseID = 1; // Пример ID склада

int quantityInWarehouse = warehouseManager.CalculateQuantityForProductInWarehouse(productID, warehouseID);

Console.WriteLine($"Количество товара с ID {productID} на складе {warehouseID}: {quantityInWarehouse}");

// Тест 3: Подсчет суммы стоимости товаров на всех складах

decimal totalValue = warehouseManager.CalculateTotalValueForProduct(productID);

Console.WriteLine($"Общая стоимость товара с ID {productID} на всех складах: {totalValue}");

// Тест 4: Подсчет суммы стоимости товаров на конкретном складе

decimal valueInWarehouse = warehouseManager.CalculateValueForProductInWarehouse(productID, warehouseID);

Console.WriteLine($"Стоимость товара с ID {productID} на складе {warehouseID}: {valueInWarehouse}");

// Тест 5: Подсчет количества товаров по категориям на всех складах

Dictionary<string, int> quantityByCategory = warehouseManager.CalculateQuantityByCategory();

Console.WriteLine("Количество товаров по категориям на всех складах:");

foreach (var category in quantityByCategory)

{

Console.WriteLine($"{category.Key}: {category.Value}");

}

// Тест 6: Подсчет количества товаров по категориям на конкретном складе

Dictionary<string, int> quantityByCategoryInWarehouse = warehouseManager.CalculateQuantityByCategoryInWarehouse(warehouseID);

Console.WriteLine($"Количество товаров по категориям на складе {warehouseID}:");

foreach (var category in quantityByCategoryInWarehouse)

{

Console.WriteLine($"{category.Key}: {category.Value}");

}

// Отрицательные тесты

// Тест 7: Подсчет общего количества для несуществующего товара

int nonExistentProductID = 999;

int totalQuantityNonExistent = warehouseManager.CalculateTotalQuantityForProduct(nonExistentProductID);

Console.WriteLine($"Общее количество несуществующего товара с ID {nonExistentProductID}: {totalQuantityNonExistent}");

// Тест 8: Подсчет количества для несуществующего товара на складе

int quantityNonExistentInWarehouse = warehouseManager.CalculateQuantityForProductInWarehouse(nonExistentProductID, warehouseID);

Console.WriteLine($"Количество несуществующего товара с ID {nonExistentProductID} на складе {warehouseID}: {quantityNonExistentInWarehouse}");

// Тест 9: Подсчет суммы стоимости для несуществующего товара

decimal totalValueNonExistent = warehouseManager.CalculateTotalValueForProduct(nonExistentProductID);

Console.WriteLine($"Общая стоимость несуществующего товара с ID {nonExistentProductID}: {totalValueNonExistent}");

// Тест 10: Подсчет суммы стоимости для несуществующего товара на складе

decimal valueNonExistentInWarehouse = warehouseManager.CalculateValueForProductInWarehouse(nonExistentProductID, warehouseID);

Console.WriteLine($"Стоимость несуществующего товара с ID {nonExistentProductID} на складе {warehouseID}: {valueNonExistentInWarehouse}");

}

}

}

Результат:

