Лабораторна робота №7-8

Горецький Максим

1. Деяке підприємство кожні 0,1 секунди виготовляє від 1 до 9 одиниць деякої продукції і передає у складське приміщення. Якщо на складі виявиться більше 50 одиниць цієї продукції, то менеджер повідомляє про можливість її одержання представникам двох підприємств споживачів, перший з яких може відразу забрати 45, а інший 48 одиниць цієї продукції. Видавати продукцію одночасно двом підприємствам немає можливості. Видача не допускається, якщо на складі залишилось менше 50 одиниць продукції, або якщо підприємство-споживач, яке першим отримало продукцію, не повернуло тару. На розвантаження продукції і повернення тари першому підприємству потрібно 1,1, а другому – 1,2 секунди. Змоделювати і проаналізувати роботу підприємства з виготовлення та видачі продукції впродовж однієї хвилини.

static int warehouse = 0;

static object lockObj = new object();

static Random rand = new Random();

static bool containerReturned = true;

static void Task1()

{

Console.WriteLine("=== ЗАВДАННЯ 1: Склад продукції ===\n");

Thread producer = new Thread(() =>

{

for (int i = 0; i < 600; i++)

{

int produced = rand.Next(1, 10);

lock (lockObj)

{

warehouse += produced;

Console.WriteLine($"[{i \* 0.1:F1}с] Виробництво: +{produced}, На складі: {warehouse}");

}

Thread.Sleep(100);

}

});

Thread consumer1 = new Thread(() =>

{

while (true)

{

lock (lockObj)

{

if (warehouse >= 50 && containerReturned)

{

warehouse -= 45;

containerReturned = false;

Console.WriteLine($" >> Споживач 1: забрав 45, залишилось {warehouse}");

Thread.Sleep(1100);

containerReturned = true;

Console.WriteLine($" >> Споживач 1: повернув тару");

}

}

Thread.Sleep(200);

}

});

Thread consumer2 = new Thread(() =>

{

while (true)

{

lock (lockObj)

{

if (warehouse >= 50 && containerReturned)

{

warehouse -= 48;

containerReturned = false;

Console.WriteLine($" >> Споживач 2: забрав 48, залишилось {warehouse}");

Thread.Sleep(1200);

containerReturned = true;

Console.WriteLine($" >> Споживач 2: повернув тару");

}

}

Thread.Sleep(200);

}

});

producer.Start();

consumer1.Start();

consumer2.Start();

Thread.Sleep(60000);

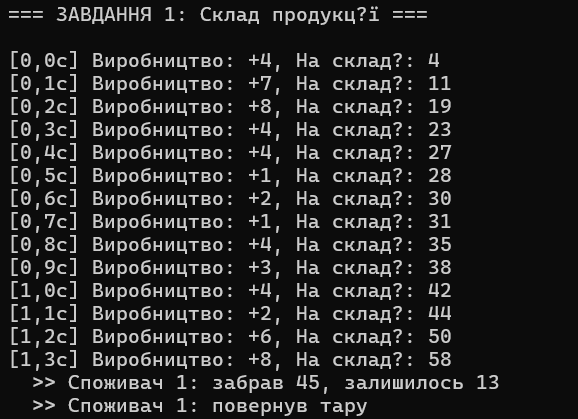
lock (lockObj)

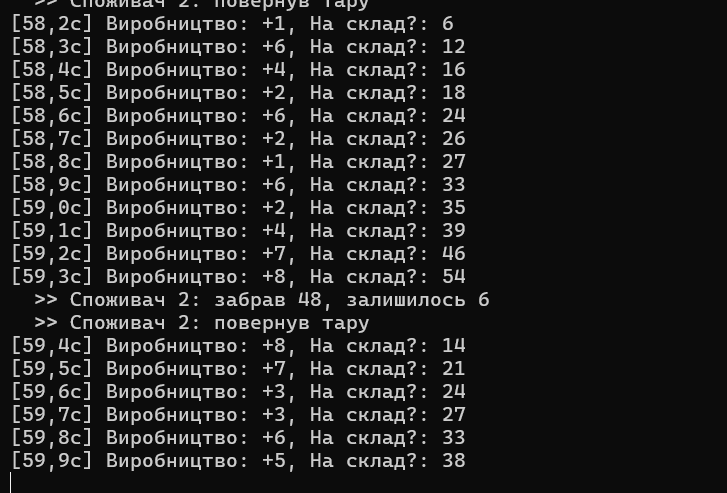
{

Console.WriteLine($"\nПідсумок: На складі залишилось {warehouse} одиниць");

}

}





2. Для реалізації 50 одиниць замовленої продукції підприємство може задіяти дві бригади пакувальників. Одна бригада, залежно від випадкових обставин, може підготувати від 10 до 60 одиниць товару, а інша від 20 до 55 одиниць. Змоделювати роботу бригад і, провівши 10 раз експерименти, з’ясувати у скількох випадках підприємство не зможе вчасно підготувати замовлену продукцію.

static void Task2()

{

Console.WriteLine("\n\n=== ЗАВДАННЯ 2: Бригади пакувальників ===\n");

int failures = 0;

for (int exp = 1; exp <= 10; exp++)

{

int packed = 0;

object lockPacked = new object();

Thread brigade1 = new Thread(() =>

{

int amount = rand.Next(10, 61);

lock (lockPacked)

{

packed += amount;

Console.WriteLine($"Експеримент {exp}: Бригада 1 упакувала {amount}");

}

});

Thread brigade2 = new Thread(() =>

{

int amount = rand.Next(20, 56);

lock (lockPacked)

{

packed += amount;

Console.WriteLine($"Експеримент {exp}: Бригада 2 упакувала {amount}");

}

});

brigade1.Start();

brigade2.Start();

brigade1.Join();

brigade2.Join();

if (packed < 50)

{

failures++;

Console.WriteLine($"Експеримент {exp}: НЕВДАЧА! Всього {packed}/50\n");

}

else

{

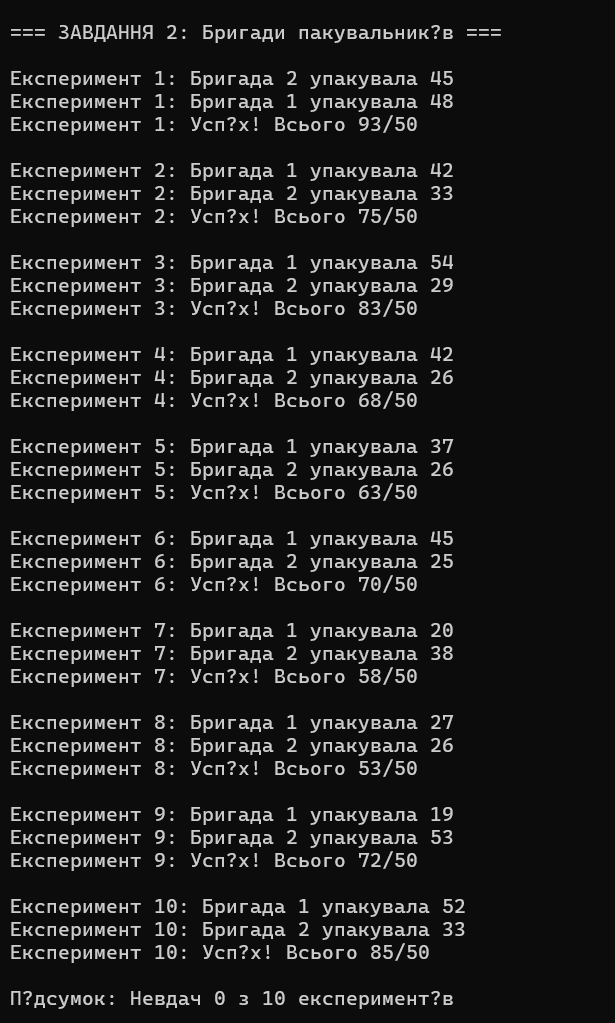
Console.WriteLine($"Експеримент {exp}: Успіх! Всього {packed}/50\n");

}

}

Console.WriteLine($"Підсумок: Невдач {failures} з 10 експериментів");

}



3. Змоделювати роботу продавця магазину з обслуговування чотирьох покупців одного і того ж виду товару, якщо кількість одиниць такого товару обмежена, об’єм покупки для кожного з покупців є випадковим значенням і можливі випадки, коли останнім у черзі покупцям не вистачить товару.

static void Task3()

{

Console.WriteLine("\n\n=== ЗАВДАННЯ 3: Магазин ===\n");

int stock = 100;

object lockStock = new object();

for (int i = 1; i <= 4; i++)

{

int customer = i;

Thread t = new Thread(() =>

{

int want = rand.Next(20, 40);

Thread.Sleep(rand.Next(100, 500));

lock (lockStock)

{

Console.WriteLine($"Покупець {customer}: хоче {want}, на складі {stock}");

if (stock >= want)

{

stock -= want;

Console.WriteLine($"Покупець {customer}: отримав {want}, залишилось {stock}");

}

else if (stock > 0)

{

Console.WriteLine($"Покупець {customer}: отримав лише {stock} (не вистачило {want - stock})");

stock = 0;

}

else

{

Console.WriteLine($"Покупець {customer}: товару немає!");

}

}

});

t.Start();

}

Thread.Sleep(2000);

Console.WriteLine($"\nПідсумок: На складі залишилось {stock} одиниць");

}

