

Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**ЛАБОРАТОРНА РОБОТА №3**

**З ДИСЦИПЛІНИ “** **ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ”**

**НА ТЕМУ: “Java. Монітори”**

**Виконав:**

Студент ІІІ курсу ФІОТ

групи ІО-82

Шендріков Євгеній

Номер у списку - 24

**Перевірив:**

Доцент Корочкін О. В.

м. Київ – 2021 р.

**Технічне завдання**

1. Розробити паралельний алгоритм рішення математичної задачі

з використанням моніторів на мові Java;

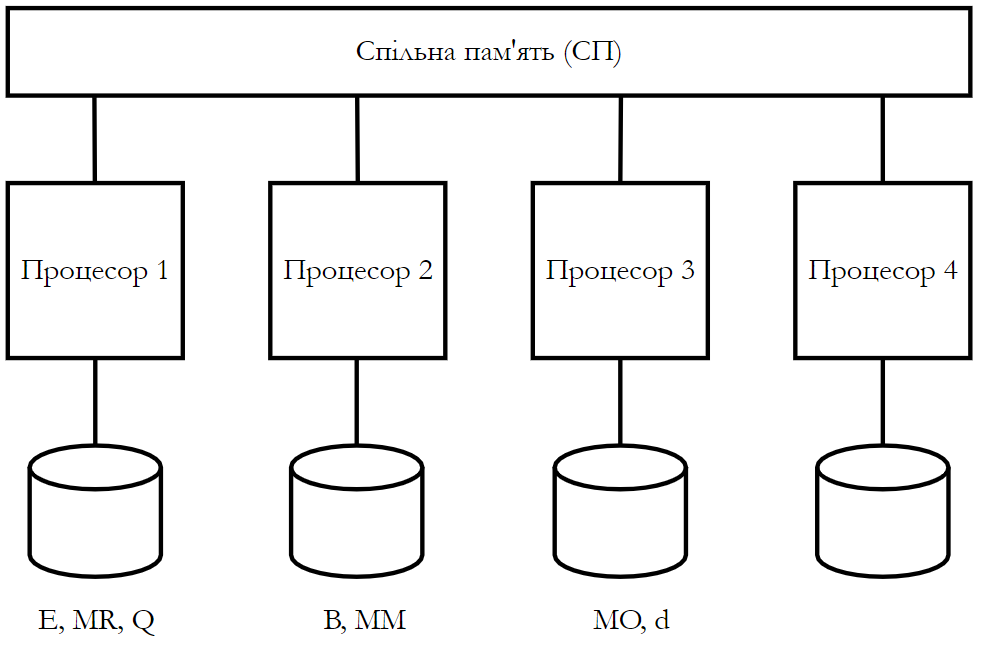
1. Виявити спільні ресурси;
2. Описати алгоритм кожного потоку (Т1 – Тр) з визначенням критичних ділянок (КД);
3. Розробити структурну схему взаємодії задач, де застосувати всі вказані засоби взаємодії процесів;
4. Розробити програму (обов’язкові “шапка”, коментарі);
5. Виконати налагодження програми;
6. Отримати правильні результати обчислень;
7. За допомогою Диспетчеру задач Windows проконтролювати завантаження ядер процесору.

Задача: ;

Мова програмування: Java;

Засоби організації взаємодії: монітори мови Java, синхронізовані блоки;

Структурна схема ПКС



**Виконання роботи**

**Етап 1. Побудова паралельного алгоритму**

1. mi = min(BH), i =
2. m = min(mi, m), i =
3. AH = (B \* MRH)
4. EH = A \* (MMH \* MO) + m \* QH \* d

Спільний ресурс: m, d, A, B, MO

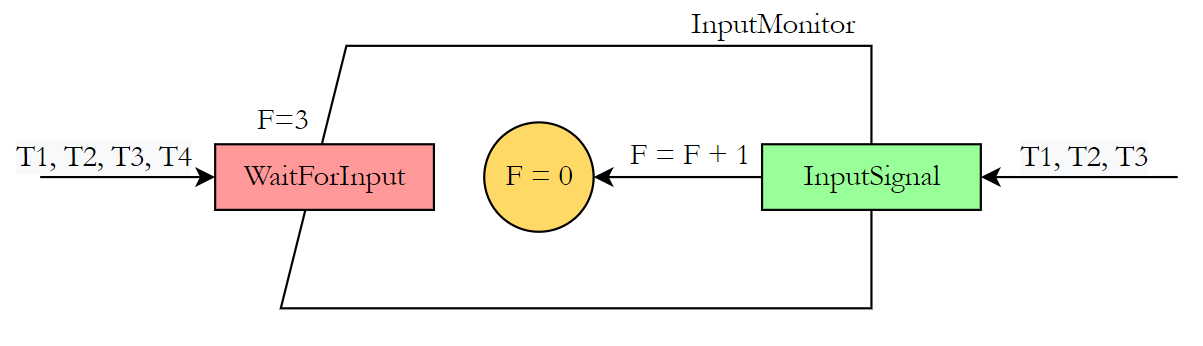
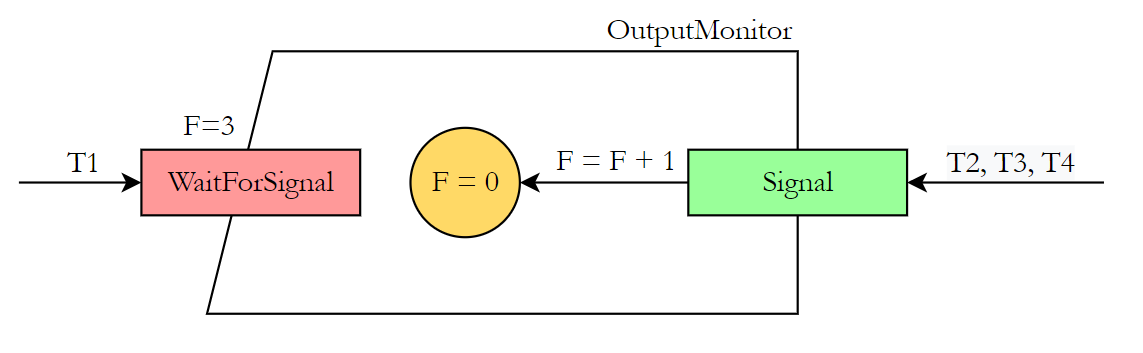
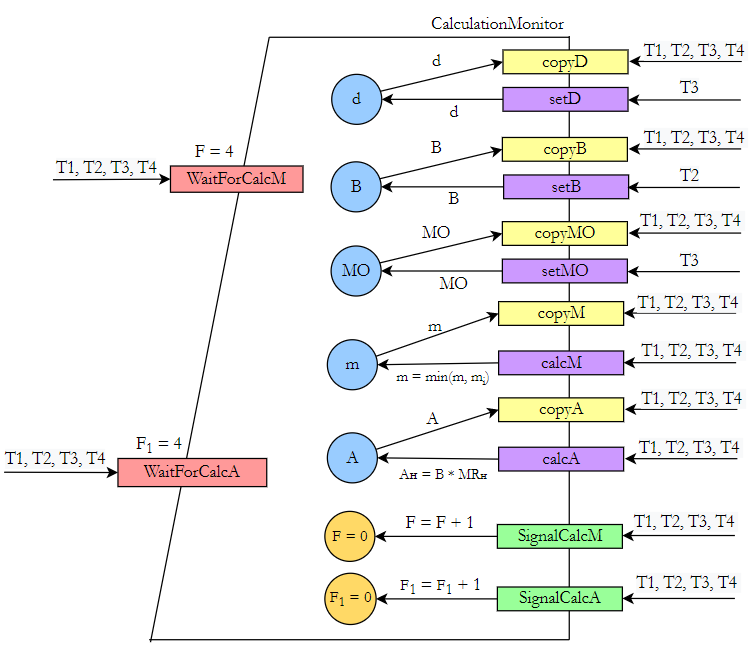
**Етап 2. Розроблення алгоритмів роботи кожного процесу**

|  |  |  |
| --- | --- | --- |
| Задача Т1 | | ТС та КД |
| 1 | Введення MR, Q |  |
| 2 | Сигнал задачам Т2, Т3, Т4 про завершення введення даних |  |
| 3 | Чекати сигналу задач T2, T3 про завершення введення даних |  |
| 4 | Копіювання B1:= B | КД1 |
| 5 | Копіювання d1:= d | КД2 |
| 6 | Копіювання MO1:= MO | КД3 |
| 7 | Обчислення m1 := min(BH) |  |
| 8 | Обчислення m: = min(m, m1) | КД4 |
| 9 | Сигнал Т2, T3, T4 про завершення обчислення m |  |
| 10 | Чекати сигнали від T2, T3, T4 про завершення обчислень m |  |
| 11 | Обчислення AH = (B1 \* MRH) | КД5 |
| 12 | Сигнал Т2, T3, T4 про завершення обчислення AH |  |
| 13 | Чекати сигнали від T2, T3, T4 про завершення обчислень AH |  |
| 14 | Копіювання m1: = m | КД6 |
| 15 | Копіювання A1:= A | КД7 |
| 16 | Обчислення EH = A1 \* (MMH \* MO1) + m1 \* QH \* d1 |  |
| 17 | Чекати на завершення обчислень E в T2, T3, T4 |  |
| 18 | Виведення E |  |
|  | | |
| Задача Т2 | | ТС та КД |
| 1 | Введення B, MM |  |
| 2 | Сигнал задачам Т1, Т3, Т4 про завершення введення даних |  |
| 3 | Чекати сигналу задач T1, T3 про завершення введення даних |  |
| 4 | Копіювання B2:= B | КД1 |
| 5 | Копіювання d2:= d | КД2 |
| 6 | Копіювання MO2:= MO | КД3 |
| 7 | Обчислення m2 := min(BH) |  |
| 8 | Обчислення m: = min(m, m2) | КД4 |
| 9 | Сигнал Т1, T3, T4 про завершення обчислення m |  |
| 10 | Чекати сигнали від T1, T3, T4 про завершення обчислень m |  |
| 11 | Обчислення AH = (B2 \* MRH) | КД5 |
| 12 | Сигнал Т1, T3, T4 про завершення обчислення AH |  |
| 13 | Чекати сигнали від T1, T3, T4 про завершення обчислень AH |  |
| 14 | Копіювання m2: = m | КД6 |
| 15 | Копіювання A2:= A | КД7 |
| 16 | Обчислення EH = A2 \* (MMH \* MO2) + m2 \* QH \* d2 |  |
| 17 | Сигнал Т1, T3, T4 про завершення обчислення E |  |
|  | | |
| Задача Т3 | | ТС та КД |
| 1 | Введення MO, d |  |
| 2 | Сигнал задачам Т1, Т2, Т4 про завершення введення даних |  |
| 3 | Чекати сигналу задач T1, T2 про завершення введення даних |  |
| 4 | Копіювання B3:= B | КД1 |
| 5 | Копіювання d3:= d | КД2 |
| 6 | Копіювання MO3:= MO | КД3 |
| 7 | Обчислення m3 := min(BH) |  |
| 8 | Обчислення m: = min(m, m3) | КД4 |
| 9 | Сигнал Т1, T2, T4 про завершення обчислення m |  |
| 10 | Чекати сигнали від T1, T2, T4 про завершення обчислень m |  |
| 11 | Обчислення AH = (B3 \* MRH) | КД5 |
| 12 | Сигнал Т1, T2, T4 про завершення обчислення AH |  |
| 13 | Чекати сигнали від T1, T2, T4 про завершення обчислень AH |  |
| 14 | Копіювання m3: = m | КД6 |
| 15 | Копіювання A3:= A | КД7 |
| 16 | Обчислення EH = A3 \* (MMH \* MO3) + m3 \* QH \* d3 |  |
| 17 | Сигнал Т1, T2, T4 про завершення обчислення E |  |
|  | | |
| Задача Т4 | | ТС та КД |
| 1 | Чекати сигналу задач T1, T2, T3 про завершення введення даних |  |
| 2 | Копіювання B4:= B | КД1 |
| 3 | Копіювання d4:= d | КД2 |
| 4 | Копіювання MO4:= MO | КД3 |
| 5 | Обчислення m4 := min(BH) |  |
| 6 | Обчислення m: = min(m, m4) | КД4 |
| 7 | Сигнал Т1, T2, T3 про завершення обчислення m |  |
| 8 | Чекати сигнали від T1, T2, T3 про завершення обчислень m |  |
| 9 | Обчислення AH = (B4 \* MRH) | КД5 |
| 10 | Сигнал Т1, T2, T3 про завершення обчислення AH |  |
| 11 | Чекати сигнали від T1, T2, T3 про завершення обчислень AH |  |
| 12 | Копіювання m4: = m | КД6 |
| 13 | Копіювання A4:= A | КД7 |
| 14 | Обчислення EH = A4 \* (MMH \* MO4) + m4 \* QH \* d4 |  |
| 15 | Сигнал Т1, T2, T3 про завершення обчислення E |  |

**Етап 3. Розроблення структурної схеми взаємодії задач**

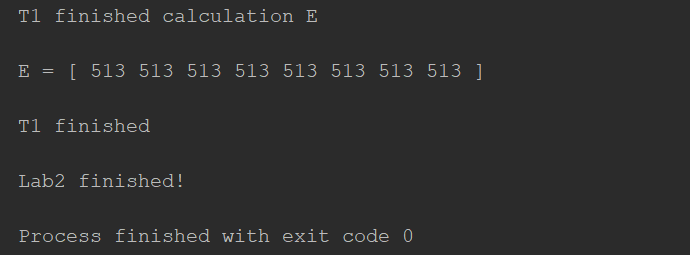
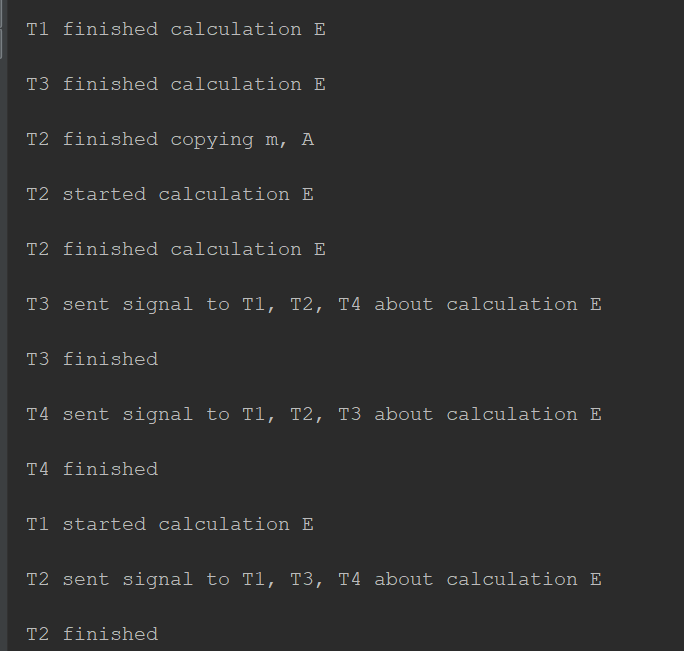
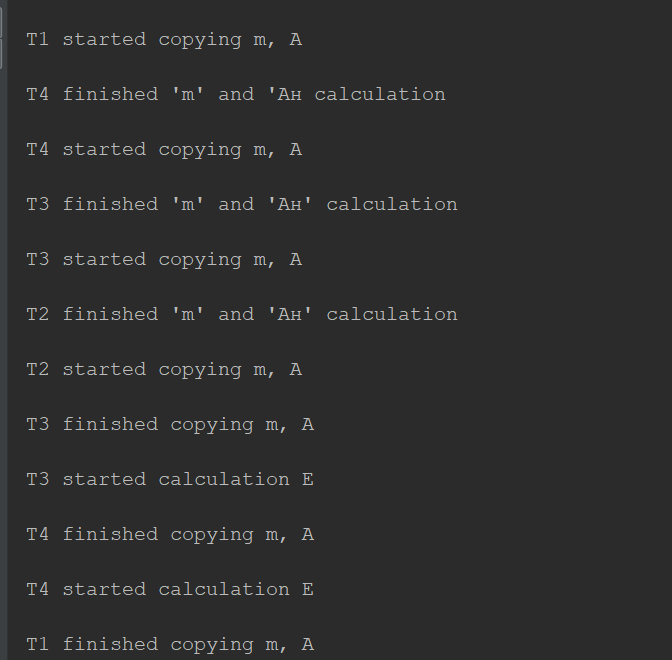
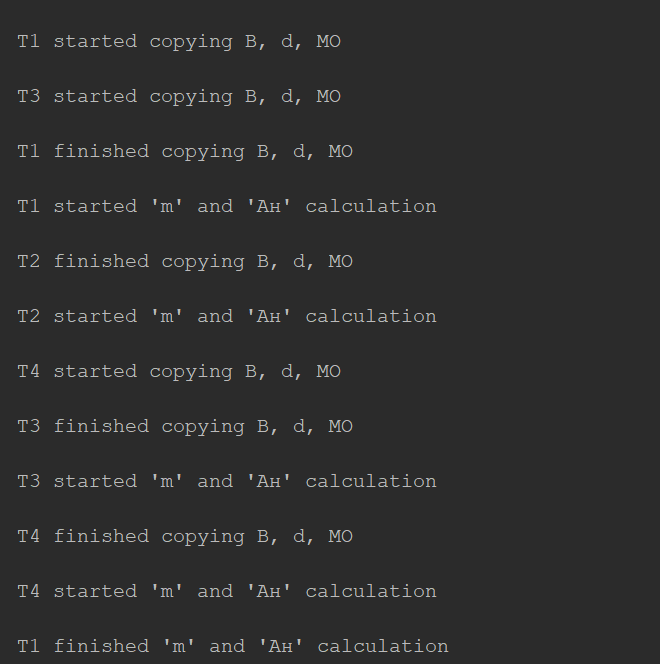
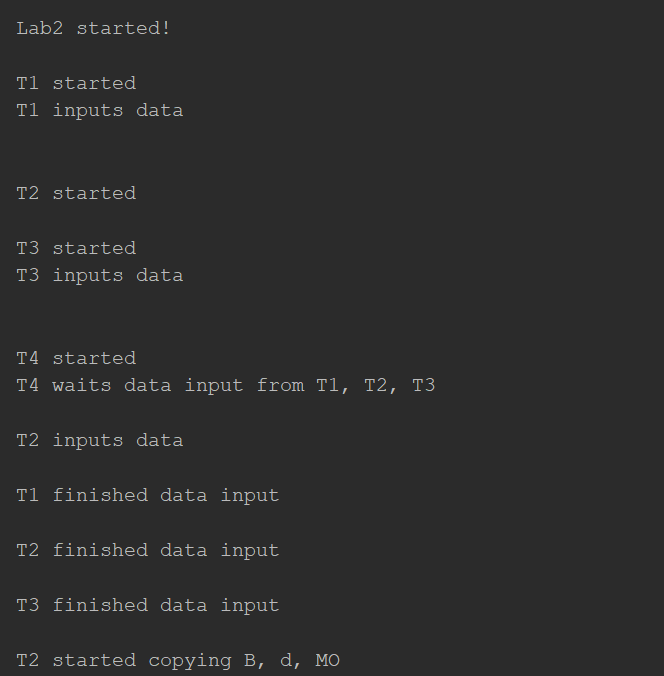
Умовні позначення на структурній схемі:

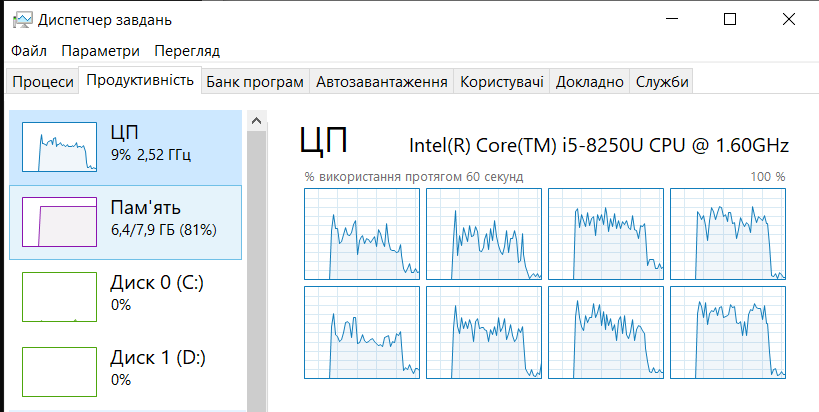
* *InputSignal* – сигнал про завершення вводу в потоках Т1, Т2, Т3
* *WaitForInput* – очікування сигналів про завершення вводу в потоках Т1, T2, Т3
* *Signal* – сигнал про завершення обчислення E у потоках Т2, Т3, Т4
* *WaitForSignal* – очікування сигналів про завершення обчислення E у потоках Т2, Т3, Т4
* *SignalCalcM* – сигнал про завершення обчислення m у потоках Т1, Т2, Т3, Т4
* *WaitForCalcM* – очікування сигналів про завершення обчислення m у потоках Т1, Т2, Т3, Т4
* *SignalCalcA* – сигнал про завершення обчислення Aн у потоках Т1, Т2, Т3, Т4
* *WaitForCalcA* – очікування сигналів про завершення обчислення Aн у потоках Т1, Т2, Т3, Т4
* *copyM* – копіювання спільного ресурсу m потоками Т1, Т2, Т3, Т4
* *calcM* – обчислення m = min(m, mi) потоками Т1, Т2, Т3, Т4
* *copyA* – копіювання спільного ресурсу A потоками Т1, Т2, Т3, Т4
* *calcA* – обчислення Aн
* *copyB* – копіювання спільного ресурсу B потоками Т1, Т2, Т3, Т4
* *setB* – введення спільного ресурсу B потоком Т2
* *copyD* – копіювання спільного ресурсу d потоками Т1, Т2, Т3, Т4
* *setD* – введення спільного ресурсу d потоком Т3
* *copyMO* – копіювання спільного ресурсу MO потоками Т1, Т2, Т3, Т4
* *setMO* – введення спільного ресурсу MO потоком Т3



**Етап 4. Розробка програми**

**Результат роботи**



****

**Висновки**

1. На основі моніторів та синхронізованих блоків мови Java було розроблено програму та паралельний алгоритм для рішення математичної задачі заданої за варіантом.
2. Було описано алгоритм кожного потоку (Т1 – Т4) з визначенням критичних ділянок (КД);
3. Розроблено структурну схему взаємодії задач, де було застосовано вказані в завданні засоби взаємодії процесів. Засобом організації взаємодії слугували монітори.
4. Було написано програму згідно з завданням та перевірено її працездатність, а також проконтрольовано завантаження ядер процесору за допомогою Диспетчеру задач. Програма забезпечує 80% завантаженості.

**Лістинг коду**

**Main.java**

1. /\*-----------------------------------------------------

2. | Labwork #3 |

3. | Java Monitors |

4. ------------------------------------------------------

5. | Author | Jack (Yevhenii) Shendrikov |

6. | Group | IO-82 |

7. | Variant | #17 |

8. | Date | 07.03.2021 |

9. ------------------------------------------------------

10. | Function | E = (B\*MR)\*(MM\*MO) + min(B)\*Q\*d |

11. ------------------------------------------------------

12. \*/

13.

14. public class Main {

15.

16. static int N = 8;

17. private static int P = 4;

18. static int H = N / P;

19.

20. static int m = Integer.MAX\_VALUE;

21. static int d;

22.

23. static int[] A = new int[N];

24. static int[] B = new int[N];

25. static int[] E = new int[N];

26. static int[] Q = new int[N];

27.

28. static int[][] MM = new int[N][N];

29. static int[][] MO = new int[N][N];

30. static int[][] MR = new int[N][N];

31.

32. public static void main(String[] args) {

33. System.out.println("\nLab2 started!");

34.

35. InputMonitor inputMonitor = new InputMonitor();

36. CalculationMonitor calcMonitor = new CalculationMonitor();

37. OutputMonitor outputMonitor = new OutputMonitor();

38.

39. T1 T1 = new T1(inputMonitor, calcMonitor, outputMonitor);

40. T2 T2 = new T2(inputMonitor, calcMonitor, outputMonitor);

41. T3 T3 = new T3(inputMonitor, calcMonitor, outputMonitor);

42. T4 T4 = new T4(inputMonitor, calcMonitor, outputMonitor);

43.

44. T1.start();

45. T2.start();

46. T3.start();

47. T4.start();

48.

49. try {

50. T1.join();

51. T2.join();

52. T3.join();

53. T4.join();

54. } catch (InterruptedException e) {

55. e.printStackTrace();

56. }

57.

58. System.out.println("\nLab2 finished!");

59. }

60. }

**T1.java**

1. import java.util.Arrays;

2.

3. public class T1 extends Thread {

4. private InputMonitor inputMonitor;

5. private CalculationMonitor calcMonitor;

6. private OutputMonitor outputMonitor;

7.

8. T1(InputMonitor inputMonitor, CalculationMonitor calcMonitor, OutputMonitor outputMonitor) {

9. this.inputMonitor = inputMonitor;

10. this.calcMonitor = calcMonitor;

11. this.outputMonitor = outputMonitor;

12. }

13.

14. @Override

15. public void run(){

16. int m1Res;

17. int start = 0, end = Main.H;

18.

19. // Початок роботи потоку Т1

20. System.out.println("\nT1 started");

21.

22. // Повідомлення про початок введення даних

23. System.out.println("T1 inputs data\n");

24.

25. // 1 - Введення вектора Q

26. Calculations.inputVector(Main.Q, 1);

27.

28. // 1 - Введення матриці MR

29. Calculations.inputMatrix(Main.MR, 1);

30.

31. // Повідомлення про завершення введення даних

32. System.out.println("T1 finished data input\n");

33.

34. // 2 - Сигнал Т2, Т3, T4 про завершення введення даних

35. inputMonitor.InputSignal();

36.

37. // 3 - Очікування завершення введення даних у інших потоках

38. inputMonitor.WaitForInput();

39.

40. // Повідомлення про початок копіювання даних у потоці Т1

41. System.out.println("T1 started copying B, d, MO\n");

42.

43. // 4 - Копія B1 = B

44. int[] B1 = calcMonitor.copyB();

45.

46. // 5 - Копія d1 = d

47. int d1 = calcMonitor.copyD();

48.

49. // 6 - Копія MO1 = MO

50. int[][] MO1 = calcMonitor.copyMO();

51.

52. // Повідомлення про закінчення копіювання даних у потоці Т1

53. System.out.println("T1 finished copying B, d, MO\n");

54.

55. // Повідомлення про початок обчислення мінімуму та вектор-матричного добутку

56. System.out.println("T1 started 'm' and 'Aн' calculation\n");

57.

58. // 7 - Обчислення m1 = min(Bн);

59. int id = 0;

60. m1Res = Calculations.vectorMin(Main.B, id); // обчислення мінімуму в 1 частині вектора В

61.

62. // 8 - Обчислення m = min(m,m1);

63. Main.m = calcMonitor.calcM(m1Res, Main.m); // обчислення мінімуму серед мінімуму

64.

65. // 9 - Сигнал потокам Т2, Т3, Т4 про закінчення обчислення m

66. calcMonitor.SignalCalcM();

67.

68. // 10 - Очікування сигналів від потоків Т2, Т3, Т4 про закінчення обчислення m

69. calcMonitor.WaitForCalcM();

70.

71. // 11 - Обчислення Aн = (B1 \* MRн)

72. for (int i = 0; i < Main.N; i++) {

73. for (int j = start; j < end; j++) {

74. Main.A[j] += B1[j] \* Main.MR[i][j];

75. }

76. }

77.

78. // 12 - Сигнал потокам Т2, Т3, Т4 про закінчення обчислення Aн

79. calcMonitor.SignalCalcA();

80.

81. // 13 - Очікування сигналів від потоків Т2, Т3, Т4 про закінчення обчислення Aн

82. calcMonitor.WaitForCalcA();

83.

84. // Повідомлення про закінчення обчислення мінімуму та вектор-матричного добутку у потоці Т1

85. System.out.println("T1 finished 'm' and 'Aн' calculation\n");

86.

87. // Повідомлення про початок копіювання даних у потоці Т1

88. System.out.println("T1 started copying m, A\n");

89.

90. // 14 - Копія m1 = m

91. int m1 = calcMonitor.copyM(Main.m);

92.

93. // 15 - Копія A1 = A

94. int[] A1 = calcMonitor.copyA();

95.

96. // Повідомлення про закінчення копіювання даних у потоці Т1

97. System.out.println("T1 finished copying m, A\n");

98.

99. // Повідомлення про початок обчислення E у потоці Т1

100. System.out.println("T1 started calculation E\n");

101.

102. // 16 - Обчислення Eн = A1 \*(MMн \* MO1) + m1 \* Qн \* d1

103. int[][] tempM = new int[Main.N][Main.N];

104. for (int j = start; j < end; j++){

105. for (int i = 0; i < Main.N; i++){

106. tempM[i][j] = 0;

107. for (int k = 0; k < Main.N; k++){

108. tempM[i][j] += MO1[i][k] \* Main.MM[j][k];

109. }

110. }

111. }

112.

113. for (int i = start; i < end; i++) {

114. int temp = 0;

115. for (int j = 0; j < Main.N; j++) {

116. temp += A1[i] \* tempM[j][i];

117. }

118. Main.E[i] = temp + m1 \* Main.Q[i] \* d1;

119. }

120.

121. // Повідомлення про закінчення обчислення E у потоці Т1

122. System.out.println("T1 finished calculation E\n");

123.

124. // 17 - Чекати на завершення обчислень E в T2, T3, T4

125. outputMonitor.WaitForSignal();

126.

127. // 18 - Виведення E

128. if (Main.N <= 15){

129. System.out.print("E = [ ");

130. Calculations.outputVector(Main.E);

131. System.out.println("]");

132. }

133. // Закінчення роботи потоку Т1

134. System.out.println("\nT1 finished");

135. }

136. }

**T2.java**

1. public class T2 extends Thread {

2. private InputMonitor inputMonitor;

3. private CalculationMonitor calcMonitor;

4. private OutputMonitor outputMonitor;

5.

6. T2(InputMonitor inputMonitor, CalculationMonitor calcMonitor, OutputMonitor outputMonitor) {

7. this.inputMonitor = inputMonitor;

8. this.calcMonitor = calcMonitor;

9. this.outputMonitor = outputMonitor;

10. }

11.

12. @Override

13. public void run(){

14. int m2Res;

15. int start = Main.H, end = Main.H \* 2;

16.

17. // Початок роботи потоку Т2

18. System.out.println("\nT2 started");

19.

20. // Повідомлення про початок введення даних

21. System.out.println("T2 inputs data\n");

22.

23. // 1 - Введення вектора B

24. Calculations.inputVector(Main.B, 1);

25.

26. // 1 - Введення матриці MM

27. Calculations.inputMatrix(Main.MM, 1);

28.

29. // Повідомлення про завершення введення даних

30. System.out.println("T2 finished data input\n");

31.

32. // 2 - Сигнал Т1, Т3, T4 про завершення введення даних

33. inputMonitor.InputSignal();

34.

35. // 3 - Очікування завершення введення даних у інших потоках

36. inputMonitor.WaitForInput();

37.

38. // Повідомлення про початок копіювання даних у потоці Т2

39. System.out.println("T2 started copying B, d, MO\n");

40.

41. // 4 - Копія B2 = B

42. int[] B2 = calcMonitor.copyB();

43.

44. // 5 - Копія d2 = d

45. int d2 = calcMonitor.copyD();

46.

47. // 6 - Копія MO2 = MO

48. int[][] MO2 = calcMonitor.copyMO();

49.

50. // Повідомлення про закінчення копіювання даних у потоці Т2

51. System.out.println("T2 finished copying B, d, MO\n");

52.

53. // Повідомлення про початок обчислення мінімуму та вектор-матричного добутку

54. System.out.println("T2 started 'm' and 'Aн' calculation\n");

55.

56. // 7 - Обчислення m2 = min(Bн);

57. int id = 1;

58. m2Res = Calculations.vectorMin(Main.B, id); // обчислення мінімуму в 2 частині вектора В

59.

60. // 8 - Обчислення m = min(m,m2);

61. Main.m = calcMonitor.calcM(m2Res, Main.m); // обчислення мінімуму серед мінімуму

62.

63. // 9 - Сигнал потокам Т1, Т3, Т4 про закінчення обчислення m

64. calcMonitor.SignalCalcM();

65.

66. // 10 - Очікування сигналів від потоків Т1, Т3, Т4 про закінчення обчислення m

67. calcMonitor.WaitForCalcM();

68.

69. // 11 - Обчислення Aн = (B2 \* MRн)

70. for (int i = 0; i < Main.N; i++) {

71. for (int j = start; j < end; j++) {

72. Main.A[j] += B2[j] \* Main.MR[i][j];

73. }

74. }

75.

76. // 12 - Сигнал потокам Т1, Т3, Т4 про закінчення обчислення Aн

77. calcMonitor.SignalCalcA();

78.

79. // 13 - Очікування сигналів від потоків Т1, Т3, Т4 про закінчення обчислення Aн

80. calcMonitor.WaitForCalcA();

81.

82. // Повідомлення про закінчення обчислення мінімуму та вектор-матричного добутку у потоці Т2

83. System.out.println("T2 finished 'm' and 'Aн' calculation\n");

84.

85. // Повідомлення про початок копіювання даних у потоці Т2

86. System.out.println("T2 started copying m, A\n");

87.

88. // 14 - Копія m2 = m

89. int m2 = calcMonitor.copyM(Main.m);

90.

91. // 15 - Копія A2 = A

92. int[] A2 = calcMonitor.copyA();

93.

94. // Повідомлення про закінчення копіювання даних у потоці Т2

95. System.out.println("T2 finished copying m, A\n");

96.

97. // Повідомлення про початок обчислення E у потоці Т2

98. System.out.println("T2 started calculation E\n");

99.

100. // 16 - Обчислення Eн = A2 \*(MMн \* MO2) + m2 \* Qн \* d2

101. int[][] tempM = new int[Main.N][Main.N];

102. for (int j = start; j < end; j++){

103. for (int i = 0; i < Main.N; i++){

104. tempM[i][j] = 0;

105. for (int k = 0; k < Main.N; k++){

106. tempM[i][j] += MO2[i][k] \* Main.MM[j][k];

107. }

108. }

109. }

110.

111. for (int i = start; i < end; i++) {

112. int temp = 0;

113. for (int j = 0; j < Main.N; j++) {

114. temp += A2[i] \* tempM[j][i];

115. }

116. Main.E[i] = temp + m2 \* Main.Q[i] \* d2;

117. }

118.

119. // Повідомлення про закінчення обчислення E у потоці Т2

120. System.out.println("T2 finished calculation E\n");

121.

122. // 17 - Сигнал Т1, T3, T4 про завершення обчислення E

123. outputMonitor.Signal();

124.

125. // Повідомлення про відправлення сигналу потокам Т1, Т3, Т4 про закінчення обчислення E

126. System.out.println("T2 sent signal to T1, T3, T4 about calculation E\n");

127.

128. // Закінчення роботи потоку Т2

129. System.out.println("T2 finished\n");

130. }

131. }

**T3.java**

1. public class T3 extends Thread {

2. private InputMonitor inputMonitor;

3. private CalculationMonitor calcMonitor;

4. private OutputMonitor outputMonitor;

5.

6. T3(InputMonitor inputMonitor, CalculationMonitor calcMonitor, OutputMonitor outputMonitor) {

7. this.inputMonitor = inputMonitor;

8. this.calcMonitor = calcMonitor;

9. this.outputMonitor = outputMonitor;

10. }

11.

12. @Override

13. public void run(){

14. int m3Res;

15. int start = Main.H \* 2, end = Main.H \* 3;

16.

17. // Початок роботи потоку Т3

18. System.out.println("\nT3 started");

19.

20. // Повідомлення про початок введення даних

21. System.out.println("T3 inputs data\n");

22.

23. // 1 - Введення числа d

24. Main.d = 1;

25.

26. // 1 - Введення матриці MO

27. Calculations.inputMatrix(Main.MO, 1);

28.

29. // Повідомлення про завершення введення даних

30. System.out.println("T3 finished data input\n");

31.

32. // 2 - Сигнал Т1, Т2, T4 про завершення введення даних

33. inputMonitor.InputSignal();

34.

35. // 3 - Очікування завершення введення даних у інших потоках

36. inputMonitor.WaitForInput();

37.

38. // Повідомлення про початок копіювання даних у потоці Т3

39. System.out.println("T3 started copying B, d, MO\n");

40.

41. // 4 - Копія B3 = B

42. int[] B3 = calcMonitor.copyB();

43.

44. // 5 - Копія d3 = d

45. int d3 = calcMonitor.copyD();

46.

47. // 6 - Копія MO3 = MO

48. int[][] MO3 = calcMonitor.copyMO();

49.

50. // Повідомлення про закінчення копіювання даних у потоці Т3

51. System.out.println("T3 finished copying B, d, MO\n");

52.

53. // Повідомлення про початок обчислення мінімуму та вектор-матричного добутку

54. System.out.println("T3 started 'm' and 'Aн' calculation\n");

55.

56. // 7 - Обчислення m3 = min(Bн);

57. int id = 2;

58. m3Res = Calculations.vectorMin(Main.B, id); // обчислення мінімуму в 3 частині вектора В

59.

60. // 8 - Обчислення m = min(m,m3);

61. Main.m = calcMonitor.calcM(m3Res, Main.m); // обчислення мінімуму серед мінімуму

62.

63. // 9 - Сигнал потокам Т1, Т2, Т4 про закінчення обчислення m

64. calcMonitor.SignalCalcM();

65.

66. // 10 - Очікування сигналів від потоків Т1, Т2, Т4 про закінчення обчислення m

67. calcMonitor.WaitForCalcM();

68.

69. // 11 - Обчислення Aн = (B3 \* MRн)

70. for (int i = 0; i < Main.N; i++) {

71. for (int j = start; j < end; j++) {

72. Main.A[j] += B3[j] \* Main.MR[i][j];

73. }

74. }

75.

76. // 12 - Сигнал потокам Т1, Т2, Т4 про закінчення обчислення Aн

77. calcMonitor.SignalCalcA();

78.

79. // 13 - Очікування сигналів від потоків Т1, Т2, Т4 про закінчення обчислення Aн

80. calcMonitor.WaitForCalcA();

81.

82. // Повідомлення про закінчення обчислення мінімуму та вектор-матричного добутку у потоці Т3

83. System.out.println("T3 finished 'm' and 'Aн' calculation\n");

84.

85. // Повідомлення про початок копіювання даних у потоці Т3

86. System.out.println("T3 started copying m, A\n");

87.

88. // 14 - Копія m3 = m

89. int m3 = calcMonitor.copyM(Main.m);

90.

91. // 15 - Копія A3 = A

92. int[] A3 = calcMonitor.copyA();

93.

94. // Повідомлення про закінчення копіювання даних у потоці Т3

95. System.out.println("T3 finished copying m, A\n");

96.

97. // Повідомлення про початок обчислення E у потоці Т3

98. System.out.println("T3 started calculation E\n");

99.

100. // 16 - Обчислення Eн = A3 \*(MMн \* MO3) + m3 \* Qн \* d3

101. int[][] tempM = new int[Main.N][Main.N];

102. for (int j = start; j < end; j++){

103. for (int i = 0; i < Main.N; i++){

104. tempM[i][j] = 0;

105. for (int k = 0; k < Main.N; k++){

106. tempM[i][j] += MO3[i][k] \* Main.MM[j][k];

107. }

108. }

109. }

110.

111. for (int i = start; i < end; i++) {

112. int temp = 0;

113. for (int j = 0; j < Main.N; j++) {

114. temp += A3[i] \* tempM[j][i];

115. }

116. Main.E[i] = temp + m3 \* Main.Q[i] \* d3;

117. }

118.

119. // Повідомлення про закінчення обчислення E у потоці Т1

120. System.out.println("T3 finished calculation E\n");

121.

122. // 17 - Сигнал Т1, T2, T4 про завершення обчислення E

123. outputMonitor.Signal();

124.

125. // Повідомлення про відправлення сигналу потокам Т1, Т2, Т4 про закінчення обчислення E

126. System.out.println("T3 sent signal to T1, T2, T4 about calculation E\n");

127.

128. // Закінчення роботи потоку Т3

129. System.out.println("T3 finished\n");

130. }

131. }

**T4.java**

1. public class T4 extends Thread {

2. private InputMonitor inputMonitor;

3. private CalculationMonitor calcMonitor;

4. private OutputMonitor outputMonitor;

5.

6. T4(InputMonitor inputMonitor, CalculationMonitor calcMonitor, OutputMonitor outputMonitor) {

7. this.inputMonitor = inputMonitor;

8. this.calcMonitor = calcMonitor;

9. this.outputMonitor = outputMonitor;

10. }

11.

12. @Override

13. public void run(){

14. int m4Res;

15. int start = Main.H \* 3, end = Main.N;

16.

17. // Початок роботи потоку Т4

18. System.out.println("\nT4 started");

19.

20. // Повідомлення про очікування завершення вводу даних у потоках Т1, Т2, Т3

21. System.out.println("T4 waits data input from T1, T2, T3\n");

22.

23. // 1 - Очікування сигналу від Т1, Т2, Т3 про завершення введення даних

24. inputMonitor.WaitForInput();

25.

26. // Повідомлення про початок копіювання даних у потоці Т4

27. System.out.println("T4 started copying B, d, MO\n");

28.

29. // 2 - Копія B4 = B

30. int[] B4 = calcMonitor.copyB();

31.

32. // 3 - Копія d4 = d

33. int d4 = calcMonitor.copyD();

34.

35. // 4 - Копія MO4 = MO

36. int[][] MO4 = calcMonitor.copyMO();

37.

38. // Повідомлення про закінчення копіювання даних у потоці Т4

39. System.out.println("T4 finished copying B, d, MO\n");

40.

41. // Повідомлення про початок обчислення мінімуму та вектор-матричного добутку

42. System.out.println("T4 started 'm' and 'Aн' calculation\n");

43.

44. // 5 - Обчислення m4 = min(Bн);

45. int id = 3;

46. m4Res = Calculations.vectorMin(Main.B, id); // обчислення мінімуму в 4 частині вектора В

47.

48. // 6 - Обчислення m = min(m,m4);

49. Main.m = calcMonitor.calcM(m4Res, Main.m); // обчислення мінімуму серед мінімуму

50.

51. // 7 - Сигнал потокам Т1, Т2, Т3 про закінчення обчислення m

52. calcMonitor.SignalCalcM();

53.

54. // 8 - Очікування сигналів від потоків Т1, Т2, Т3 про закінчення обчислення m

55. calcMonitor.WaitForCalcM();

56.

57. // 9 - Обчислення Aн = (B4 \* MRн)

58. for (int i = 0; i < Main.N; i++) {

59. for (int j = start; j < end; j++) {

60. Main.A[j] += B4[j] \* Main.MR[i][j];

61. }

62. }

63.

64. // 10 - Сигнал потокам Т1, Т2, Т3 про закінчення обчислення Aн

65. calcMonitor.SignalCalcA();

66.

67. // 11 - Очікування сигналів від потоків Т1, Т2, Т3 про закінчення обчислення Aн

68. calcMonitor.WaitForCalcA();

69.

70. // Повідомлення про закінчення обчислення мінімуму та вектор-матричного добутку у потоці Т4

71. System.out.println("T4 finished 'm' and 'Aн calculation\n");

72.

73. // Повідомлення про початок копіювання даних у потоці Т4

74. System.out.println("T4 started copying m, A\n");

75.

76. // 12 - Копія m4 = m

77. int m4 = calcMonitor.copyM(Main.m);

78.

79. // 13 - Копія A4 = A

80. int[] A4 = calcMonitor.copyA();

81.

82. // Повідомлення про закінчення копіювання даних у потоці Т4

83. System.out.println("T4 finished copying m, A\n");

84.

85. // Повідомлення про початок обчислення E у потоці Т4

86. System.out.println("T4 started calculation E\n");

87.

88. // 14 - Обчислення Eн = A4 \*(MMн \* MO4) + m4 \* Qн \* d4

89. int[][] tempM = new int[Main.N][Main.N];

90. for (int j = start; j < end; j++){

91. for (int i = 0; i < Main.N; i++){

92. tempM[i][j] = 0;

93. for (int k = 0; k < Main.N; k++){

94. tempM[i][j] += MO4[i][k] \* Main.MM[j][k];

95. }

96. }

97. }

98.

99. for (int i = start; i < end; i++) {

100. int temp = 0;

101. for (int j = 0; j < Main.N; j++) {

102. temp += A4[i] \* tempM[j][i];

103. }

104. Main.E[i] = temp + m4 \* Main.Q[i] \* d4;

105. }

106.

107. // Повідомлення про закінчення обчислення E у потоці Т1

108. System.out.println("T1 finished calculation E\n");

109.

110. // 15 - Сигнал Т1, T2, T4 про завершення обчислення E

111. outputMonitor.Signal();

112.

113. // Повідомлення про відправлення сигналу потокам Т1, Т2, Т3 про закінчення обчислення E

114. System.out.println("T4 sent signal to T1, T2, T3 about calculation E\n");

115.

116. // Закінчення роботи потоку Т4

117. System.out.println("T4 finished\n");

118. }

119. }

**InputMonitor.java**

1. class InputMonitor {

2. private static int F = 0;

3.

4. synchronized void InputSignal() {

5. F++;

6. if (F >= 3) notifyAll();

7. }

8.

9. synchronized void WaitForInput() {

10. try {

11. if (F < 2) wait();

12. } catch(Exception e){

13. e.printStackTrace();

14. }

15. }

16. }

**OutputMonitor.java**

1. class OutputMonitor {

2. private static int F = 0;

3.

4. synchronized void Signal() {

5. F++;

6. if (F >= 3) notifyAll();

7. }

8.

9. synchronized void WaitForSignal() {

10. try {

11. if (F < 3) wait();

12. } catch(Exception e){

13. e.printStackTrace();

14. }

15. }

16. }

**CalculationMonitor.java**

1. class CalculationMonitor {

2. private static int F = 0;

3. private static int F1 = 0;

4. private int d = Main.d;

5. private int[] A = Main.A;

6. private int[] B = Main.B;

7. private int[][] MO = Main.MO;

8.

9. synchronized int copyM(int x) {

10. return x;

11. }

12.

13. synchronized int calcM(int val1, int val2) { return val1 < val2 ? val1 : val2; }

14.

15. synchronized int[] copyA() {

16. return this.A;

17. }

18.

19. synchronized int copyD() { return this.d; }

20.

21. synchronized int[] copyB() { return this.B; }

22.

23. synchronized int[][] copyMO() { return this.MO; }

24.

25. synchronized void SignalCalcM() {

26. F++;

27. if (F >= 4) notifyAll();

28. }

29.

30. synchronized void SignalCalcA() {

31. F1++;

32. if (F1 >= 4) notifyAll();

33. }

34.

35. synchronized void WaitForCalcM() {

36. try {

37. if (F < 4) wait();

38. } catch(Exception e){

39. e.printStackTrace();

40. }

41. }

42. synchronized void WaitForCalcA() {

43. try {

44. if (F1 < 4) wait();

45. } catch(Exception e){

46. e.printStackTrace();

47. }

48. }

49. }

**Data.java**

1. class Calculations {

2. private static int H = Main.H;

3.

4. public static void inputVector(int[] Vect, int val){

5. for (int i = 0; i < Vect.length; i++){

6. Vect[i] = val;

7. }

8. }

9.

10. public static void inputMatrix(int[][] Matr, int val){

11. for (int i = 0; i < Matr.length; i++)

12. for (int j = 0; j < Matr[i].length; j++)

13. Matr[i][j] = val;

14. }

15.

16. public static void outputVector(int[] Vect){

17. for (int aVect : Vect) {

18. System.out.print(aVect + " ");

19. }

20. }

21. public static int vectorMin(int[] vector, int id)

22. {

23. int min = vector[H \* id];

24. for (int i = H \* id; i < H \* (id + 1); i++) {

25. if (vector[i] < min)

26. min = vector[i];

27. }

28. return min;

29. }

30. }