УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра программного обеспечения информационных технологий

Отчет

По лабораторной работе №4

по дисциплине

Системный анализ и машинное моделирование

вариант 11

Выполнил: Студент гр. 851006

Петровец В.Н.

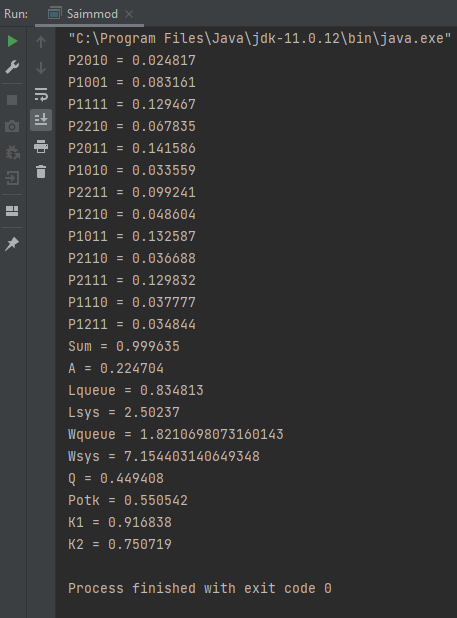
Проверил: Мельник Н. И.

Минск 2021

1. **Листинг кода программы**

1. package by.petrovlad;

1. import java.util.Random;
2. import java.util.Scanner;
3. public class Saimmod {
4. private static final int N = 1000000;
5. private static final String INITIAL\_STATE = "2000";
6. public static void main(String[] args) {
7. int P2000 = 0, P1000 = 0, P2010 = 0,
8. P1010 = 0, P2110 = 0, P1110 = 0,
9. P1001 = 0, P2011 = 0, P1011 = 0,
10. P2111 = 0, P1111 = 0, P2210 = 0,
11. P1210 = 0, P2211 = 0, P1211 = 0;
12. int firstChannel = 0, secondChannel = 0,
13. queueLength = 0, requestLength = 0,
14. processedCount = 0, generatedCount = 0,
15. declinedCount = 0;
16. String state = INITIAL\_STATE;
17. double pi1, pi2;
18. Scanner scanner = new Scanner(System.in);
19. /\* System.out.print("Введите pi1: ");
20. pi1 = scanner.nextDouble();
21. System.out.print("Введите pi2: ");
22. pi2 = scanner.nextDouble();\*/
23. pi1 = 0.5d;
24. pi2 = 0.7d;
25. int processed1 = 0;
26. int l1 = 0;
27. int l2 = 0;
28. Random rand = new Random();
29. for (int i = 0; i < N; i++) {
30. double currPi1 = rand.nextDouble();
31. double currPi2 = rand.nextDouble();
32. boolean isProcessed1 = isProcessed(currPi1, pi1);
33. boolean isProcessed2 = isProcessed(currPi2, pi2);
34. switch (state) {
35. case "2000":
36. P2000++;
37. state = "1000";
38. break;
39. case "1000":
40. P1000++;
41. state = "2010";
42. break;
43. case "2010":
44. P2010++;
45. state = isProcessed1 ? "1001" : "1010";
46. if (isProcessed1) processed1++;
47. break;
48. case "1010":
49. P1010++;
50. state = isProcessed1 ? "2011" : "2110";
51. if (isProcessed1) processed1++;
52. break;
53. case "2110":
54. P2110++;
55. state = isProcessed1 ? "1011" : "1110";
56. if (isProcessed1) processed1++;
57. break;
58. case "1110":
59. P1110++;
60. state = isProcessed1 ? "2111" : "2210";
61. if (isProcessed1) processed1++;
62. break;
63. case "2210":
64. P2210++;
65. state = isProcessed1 ? "1111" : "1210";
66. if (isProcessed1) processed1++;
67. break;
68. case "1210":
69. P1210++;
70. state = isProcessed1 ? "2211" : "2210";
71. if (!isProcessed1) {
72. declinedCount++;
73. }
74. if (isProcessed1) processed1++;
75. break;
76. case "1001":
77. P1001++;
78. state = isProcessed2 ? "2010" : "2011";
79. if (isProcessed2) processedCount++;
80. break;
81. case "2011":
82. P2011++;
83. if (isProcessed1 && isProcessed2) {
84. state = "1001";
85. processedCount++;
86. }
87. if (!isProcessed1 && isProcessed2) {
88. state = "1010";
89. processedCount++;
90. }
91. if (isProcessed1 && !isProcessed2) {
92. state = "1001";
93. declinedCount++;
94. }
95. if (!isProcessed1 && !isProcessed2) {
96. state = "1011";
97. }
98. if (isProcessed1) processed1++;
99. break;
100. case "1011":
101. P1011++;
102. if (isProcessed1 && isProcessed2) {
103. state = "2011";
104. processedCount++;
105. }
106. if (!isProcessed1 && isProcessed2) {
107. state = "2110";
108. processedCount++;
109. }
110. if (isProcessed1 && !isProcessed2) {
111. state = "2011";
112. declinedCount++;
113. }
114. if (!isProcessed1 && !isProcessed2) {
115. state = "2111";
116. }
117. if (isProcessed1) processed1++;
118. break;
119. case "2111":
120. P2111++;
121. if (isProcessed1 && isProcessed2) {
122. state = "1011";
123. processedCount++;
124. }
125. if (!isProcessed1 && isProcessed2) {
126. state = "1110";
127. processedCount++;
128. }
129. if (isProcessed1 && !isProcessed2) {
130. state = "1011";
131. declinedCount++;
132. }
133. if (!isProcessed1 && !isProcessed2) {
134. state = "1111";
135. }
136. if (isProcessed1) processed1++;
137. break;
138. case "1111":
139. P1111++;
140. if (isProcessed1 && isProcessed2) {
141. state = "2111";
142. processedCount++;
143. }
144. if (!isProcessed1 && isProcessed2) {
145. state = "2210";
146. processedCount++;
147. }
148. if (isProcessed1 && !isProcessed2) {
149. state = "2111";
150. declinedCount++;
151. }
152. if (!isProcessed1 && !isProcessed2) {
153. state = "2211";
154. }
155. if (isProcessed1) processed1++;
156. break;
157. case "2211":
158. P2211++;
159. if (isProcessed1 && isProcessed2) {
160. state = "1111";
161. processedCount++;
162. }
163. if (!isProcessed1 && isProcessed2) {
164. state = "1210";
165. processedCount++;
166. }
167. if (isProcessed1 && !isProcessed2) {
168. state = "1111";
169. declinedCount++;
170. }
171. if (!isProcessed1 && !isProcessed2) {
172. state = "1211";
173. }
174. if (isProcessed1) processed1++;
175. break;
176. case "1211":
177. P1211++;
178. if (!isProcessed1 && isProcessed2) {
179. state = "2210";
180. processedCount++;
181. } else {
182. state = "2211";
183. declinedCount++;
184. if (isProcessed2) {
185. processedCount++;
186. }
187. }
188. if (isProcessed1) processed1++;
189. break;
190. }
191. if (state.charAt(0) == '1') generatedCount++;
192. int J = Integer.parseInt(String.valueOf(state.charAt(1)));
193. int T1 = Integer.parseInt(String.valueOf(state.charAt(2)));
194. int T2 = Integer.parseInt(String.valueOf(state.charAt(3)));
195. queueLength += J;
196. firstChannel += T1;
197. secondChannel += T2;
198. requestLength += J + T1 + T2;
199. l1 += J + T1;
200. l2 += T2;
201. }
202. System.out.println("P2010 = " + (double)P2010 / N);
203. System.out.println("P1001 = " + (double)P1001 / N);
204. System.out.println("P1111 = " + (double)P1111 / N);
205. System.out.println("P2210 = " + (double)P2210 / N);
206. System.out.println("P2011 = " + (double)P2011 / N);
207. System.out.println("P1010 = " + (double)P1010 / N);
208. System.out.println("P2211 = " + (double)P2211 / N);
209. System.out.println("P1210 = " + (double)P1210 / N);
210. System.out.println("P1011 = " + (double)P1011 / N);
211. System.out.println("P2110 = " + (double)P2110 / N);
212. System.out.println("P2111 = " + (double)P2111 / N);
213. System.out.println("P1110 = " + (double)P1110 / N);
214. System.out.println("P1211 = " + (double)P1211 / N);
215. System.out.println("Сумма = " + (double)(P2000 + P1000 + P2010 + P1001 + P1111 + P2210 + P2011 + P1010
216. + P2211 + P1210 + P1011 + P2110 + P1111 + P1110 + P1211) / N);
217. System.out.println("A = " + (double)processedCount / N);
218. System.out.println("Lоч = " + (double)queueLength / N);
219. System.out.println("Lс = " + (double)requestLength / N);
220. System.out.println("Wqueue = " + (double)queueLength / processed1);
221. //System.out.println("Wоч = " + (double)queueLength / ((1 - pi1) \* (firstChannel)));
222. //System.out.println("Wsys = " + ((double)queueLength / ((1 - pi1) \* (firstChannel)) + 1 / (1 - pi1) + 1 / (1 - pi2)));
223. //System.out.println("Wc = " + ((double)l1 / processedCount + l2 / processedCount));
224. System.out.println("Wc = " + ((double)l1 / processed1 + (double)l2 / processedCount));
225. System.out.println("Q = " + (double)processedCount / generatedCount);
226. System.out.println("Pотк = " + (double)declinedCount / generatedCount);
227. System.out.println("K1 = " + (double)firstChannel / N);
228. System.out.println("K2 = " + (double)secondChannel / N);
229. }
230. private static boolean isProcessed(double currentPi, double pi) {
231. return currentPi > pi;
232. }
233. }

**2. Результат выполнения программы**