

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра прикладной математики

## КУРСОВАЯ РАБОТА

по экономико-математическим методам и моделям

Обучающая программа по теме

«Замена оборудования»

Студент

\_\_\_\_\_  
подпись, дата

Суханов Д.И.  
фамилия, инициалы

Группа

ПМ-19-1

Руководитель

\_\_\_\_\_  
ученая степень, ученое звание

\_\_\_\_\_  
подпись, дата

Лубенец Ю.В.  
фамилия, инициалы

Липецк 2022 г.

# Содержание

Цель работы . . . . .	3
Теоретическая часть . . . . .	4
Задача замены оборудования . . . . .	4
Алгоритм решения задачи замены оборудования . . . . .	4
Практическая часть . . . . .	5
Памятка пользователя . . . . .	5
Фрагменты листинга программы . . . . .	12
Заключение . . . . .	25
Список литературы . . . . .	26

## Цель работы

Создание программного продукта для решения задачи замены оборудования.

# Теоретическая часть

## Задача замены оборудования

Задача о замене оборудования рассматривается в различных формулировках. Более часто рассматриваются задачи о замене оборудования — задачи о минимизации расходов.

Пусть заданы функции:  $r_k(t)$  - затраты на эксплуатацию оборудования возраста  $t$  в течение  $k$ -го месяца,  $p_k$ , — стоимость нового оборудования в начале  $k$ -го месяца,  $\varphi_k(t)$  — ликвидная Стоимость оборудования возраста  $t$  в течение  $k$ -го месяца.

Кроме того, будем полагать, что в начале периода оборудование является новым, а в конце периода оборудование продается по ликвидной стоимости.

Требуется найти план эксплуатации оборудования, при котором затраты за весь период минимальны.

## Алгоритм решения задачи замены оборудования

Обозначим  $u$  - управление в этой задаче, которое может принимать два значения:  $u_0$  — не делается замена оборудования,  $u_1$  — делается замена оборудования.

Обозначим  $P_k(t)$  — минимальные затраты на эксплуатацию оборудования с  $k$ -го месяца до конца периода, если в начале  $k$ -го месяца возраст оборудования равен  $t$ .

Рекуррентные соотношения имеют следующий вид:

Для последнего месяца:

$$P_n(t) = \min \begin{cases} r_n(t) - \varphi_{n+1}(t+1) & u = u_0 \\ p_n - \varphi_n(t) + r_n(0) - \varphi_{n+1}(1) & u = u_1 \end{cases}$$

Для остальных месяцев:

$$P_k(t) = \min \begin{cases} r_k(t) - P_{k+1}(t+1) & u = u_0 \\ p_k - \varphi_k(t) + r_k(0) - P_{k+1}(1) & u = u_1 \end{cases}$$

Обратной условной оптимизацией находим  $P_n(t), P_{n-1}(t), \dots, P_1(t)$ , затем прямой оптимизацией — план замены оборудования.

# Практическая часть

## Памятка пользователя

После запуска программы нам открывается главное окно программы.

Задача о замене оборудования

Режим решения Alt + F Справка F4 Выход F5

**Начальные данные**

$n = 4, \varphi_k(t) = p_k \cdot 2^{-t}, r_k(t) = 0, 2p_k(t+1) n = 4,$

$p_k =$    $+$    $(k-1)$

k	t	u=u0	u=u1	u	Pk(t)
4	1	<input type="text"/> - <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
	2	<input type="text"/> - <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
	3	<input type="text"/> - <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
3	1	<input type="text"/> + <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
	2	<input type="text"/> + <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
2	1	<input type="text"/> + <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
1	0	<input type="text"/> + <input type="text"/> = 0	-----	u0	0

Рисунок 1. Главное окно программы.

Перед началом работы следует выбрать режим решения. При выборе ручного решения программа предоставит выбор между вводом своих данных и данных из файла.

Задача о замене оборудования

Режим решения Alt + F Справка F4 Выход F5

Ручной G Данные из файла F1

Рассчитать автоматически (off) F3 Свои данные F2

**Начальные данные**

$n = 4, \varphi_k(t) = p_k \cdot 2^{-t}, r_k(t) = 0, 2p_k(t+1) n = 4,$

$p_k =$    $+$    $(k-1)$

k	t	u=u0	u=u1	u	Pk(t)
4	1	<input type="text"/> - <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
	2	<input type="text"/> - <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
	3	<input type="text"/> - <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
3	1	<input type="text"/> + <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
	2	<input type="text"/> + <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
2	1	<input type="text"/> + <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
1	0	<input type="text"/> + <input type="text"/> = 0	-----	u0	0

Рисунок 2. Выбор режима решения.

При выборе ручного режима решения задачи на данных из файла программа предложит ввести номер варианта.

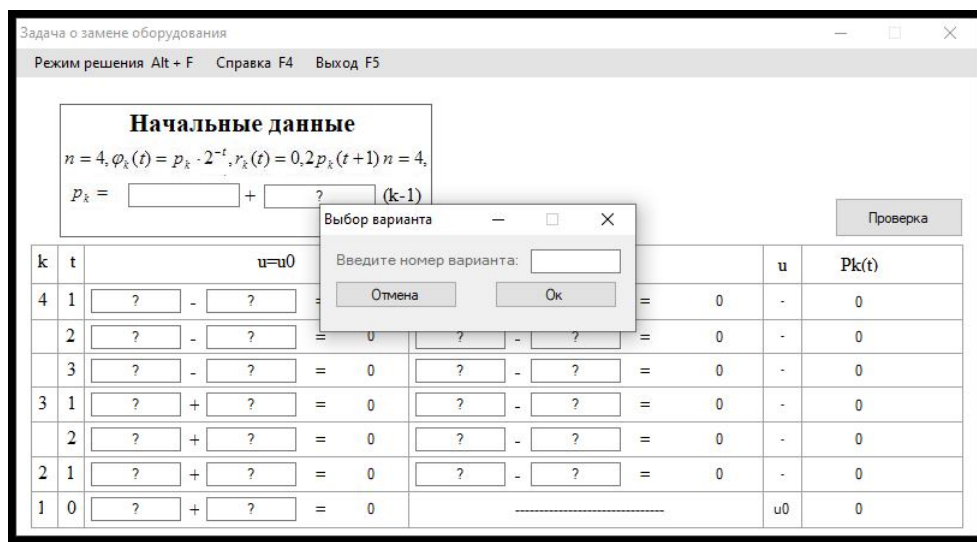


Рисунок 3. Выбор варианта.

Если при проверке решения программа найдет ошибки, то она оповестит об этом пользователя диалоговым окном и выделит все неправильные значения главной таблицы.

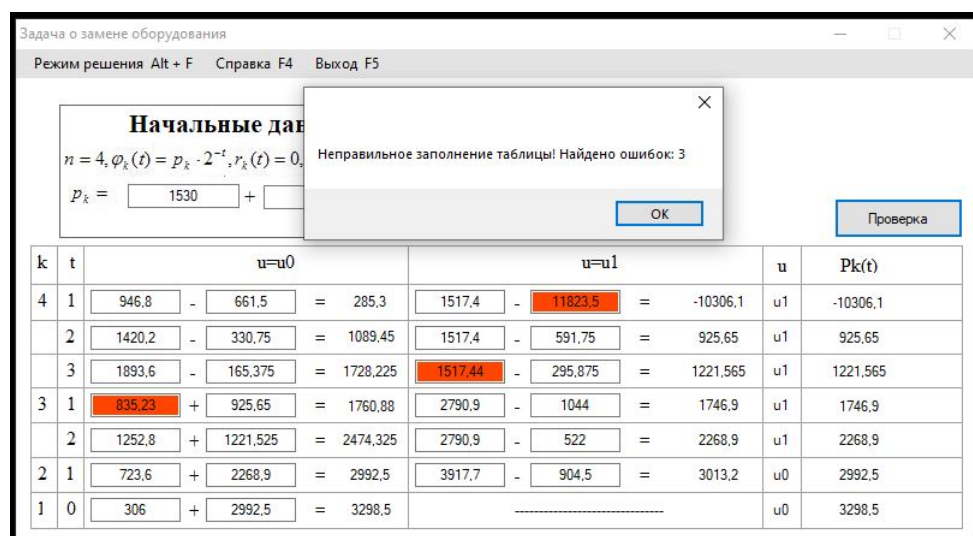


Рисунок 4. Ошибки на главном экране.

После успешного заполнения главной таблицы программа предоставит возможность заполнить таблицу прямого планирования.

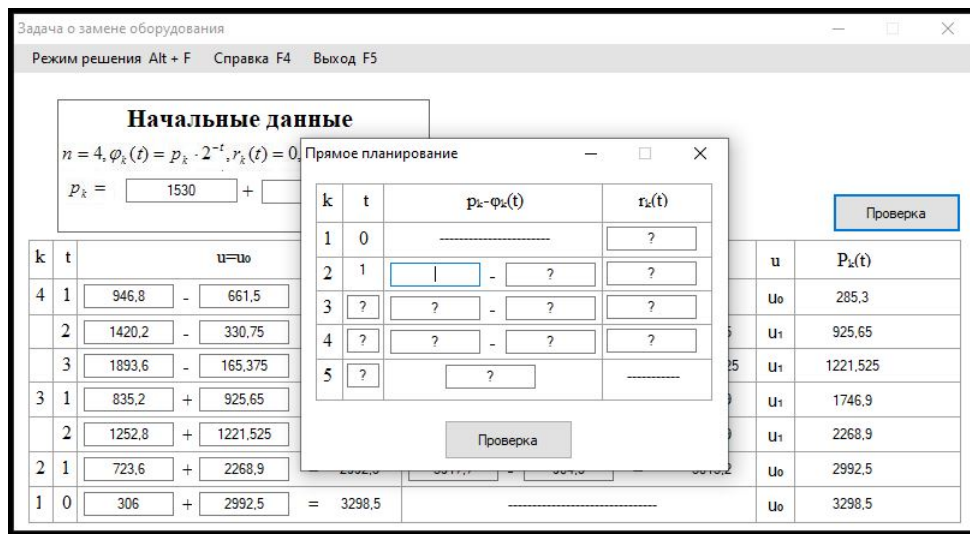


Рисунок 5. Прямое планирование.

Если будут допущены ошибки, то программа оповестит пользователя диалоговым окном и подсветит неправильные значения в таблице.



Рисунок 6. Ошибки в таблице прямого планирования.

После правильного заполнения всех таблиц программа выдаст сообщение об успешном завершении решения задачи.

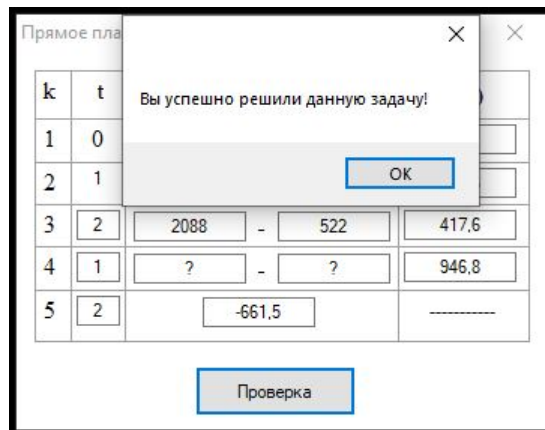


Рисунок 7. Успешное решение.

Программа предоставляет возможность автоматического решения задачи. При включенном режиме таблицы будут автоматически заполняться при проверке.

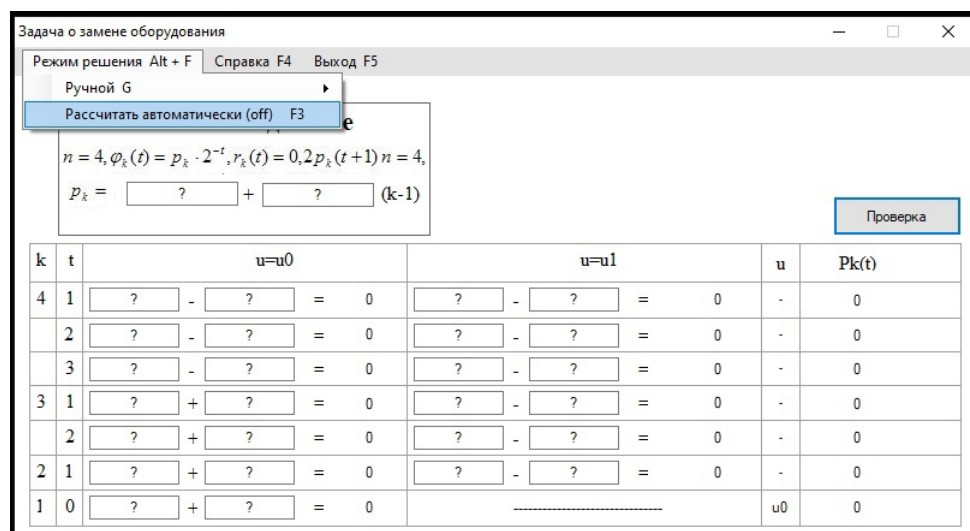


Рисунок 8. Выбор автоматического режима.



Для активации режима автоматического расчета необходимо ввести пароль.

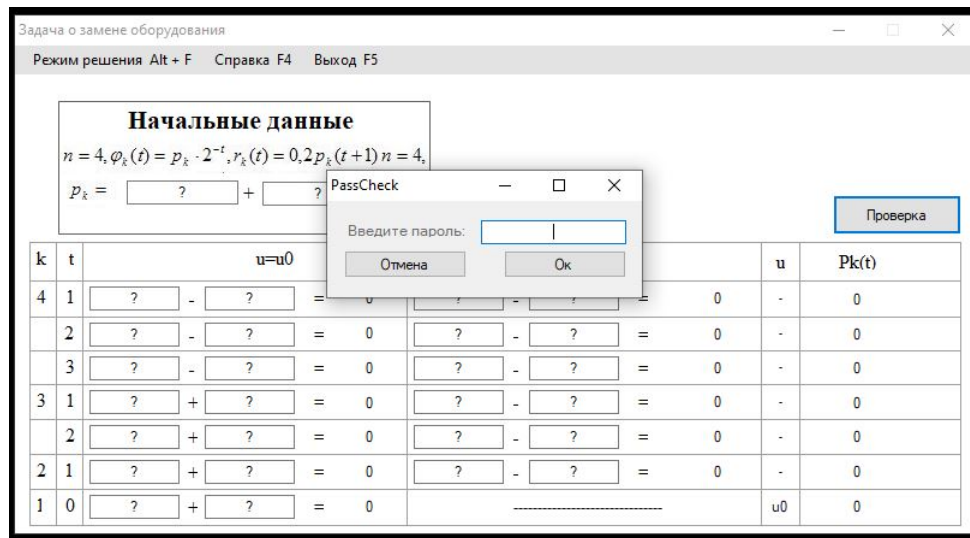


Рисунок 9. Ограниченный доступ к режиму.

При успешном включении режима автоматического расчета, программа оповестит пользователя сообщением.

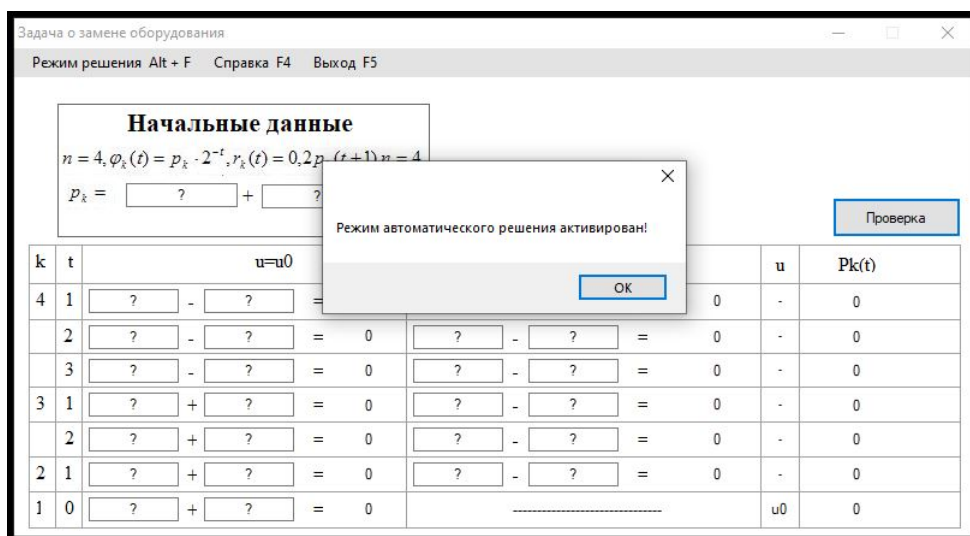


Рисунок 10. Успешная активация режима.

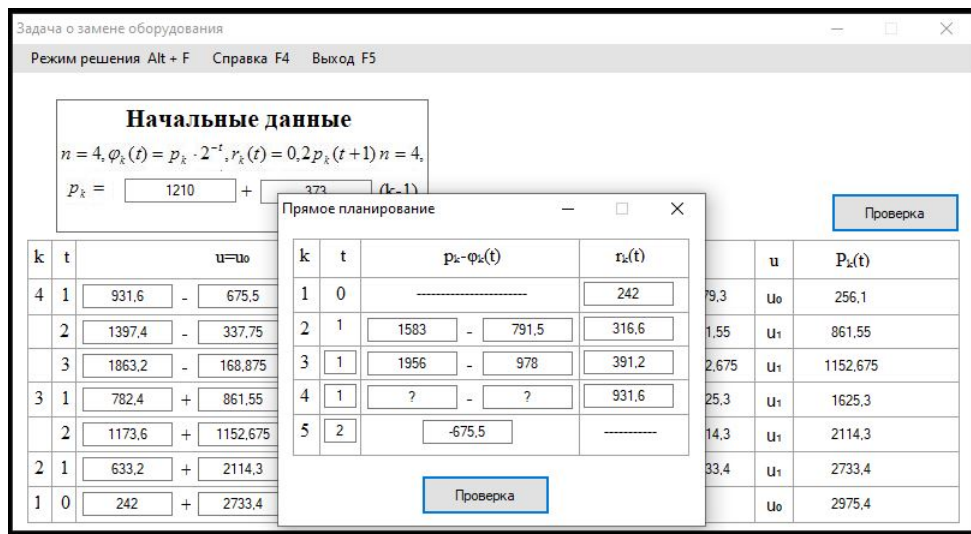


Рисунок 11. Пример автоматического режима.

Для того чтобы выключить автоматический режим необходимо повторно выбрать данный режим.

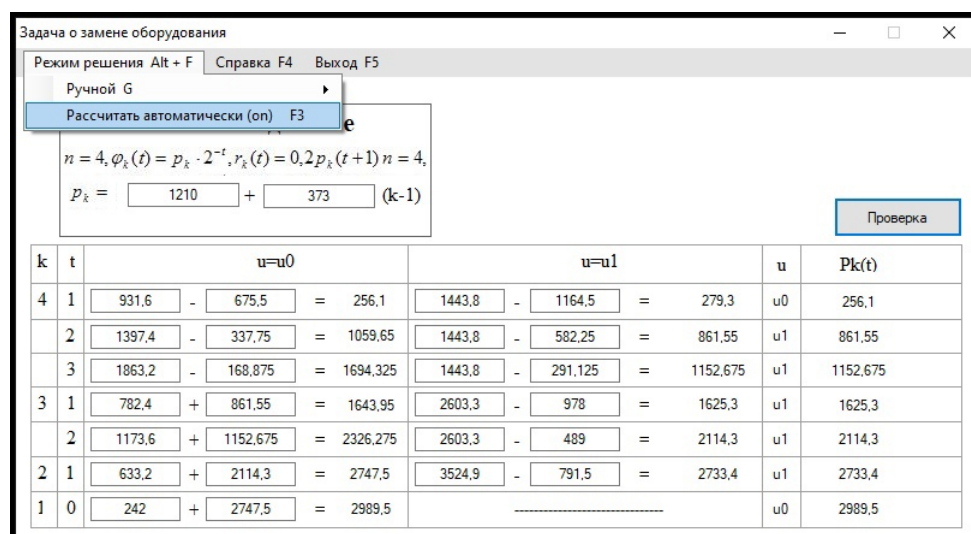


Рисунок 12. Выключение автоматического режима.

При выборе в пункте меню пункта «справка» программа откроет pdf инструкцию по использованию программы.

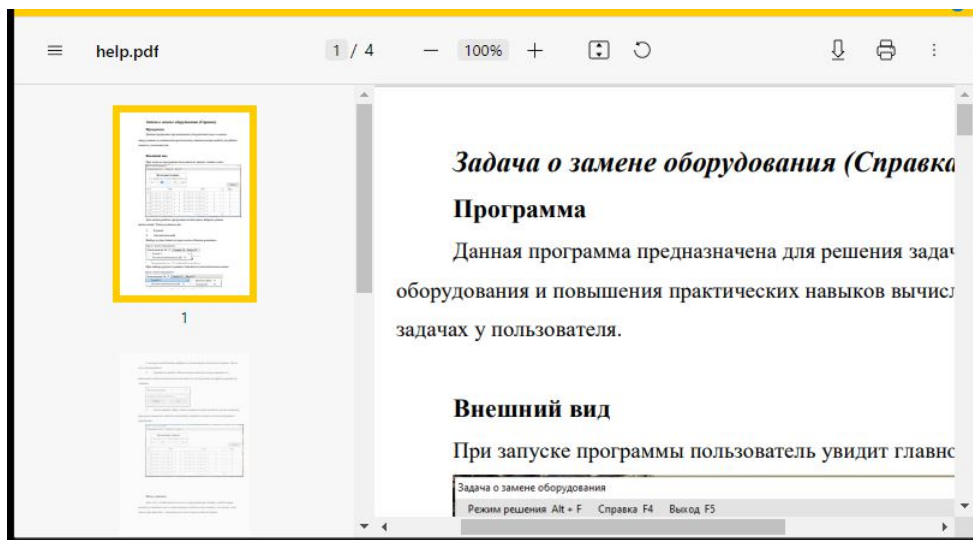


Рисунок 13. Инструкция по использованию.

The screenshot shows the main window of the 'Task on equipment replacement' program. The title bar is 'Задача о замене оборудования'. The menu bar has 'Режим решения Alt + F', 'Справка F4', and 'Выход F5'. The main content area has a section titled 'Начальные данные' with the following text:  $n = 4, \varphi_k(t) = p_k \cdot 2^{-t}, r_k(t) = 0, 2p_k(t+1) n = 4,$  and  $p_k = \text{?} + \text{?} (k-1)$ . Below this is a table with columns 'k', 't', 'u=u0', 'u=u1', 'u', and 'Pk(t)'. The table contains several rows of input fields and calculations. A 'Проверка' button is located to the right of the table.

k	t	u=u0	u=u1	u	Pk(t)
4	1	<input type="text"/> - <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
	2	<input type="text"/> - <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
	3	<input type="text"/> - <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
3	1	<input type="text"/> + <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
	2	<input type="text"/> + <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
2	1	<input type="text"/> + <input type="text"/> = 0	<input type="text"/> - <input type="text"/> = 0	-	0
1	0	<input type="text"/> + <input type="text"/> = 0	-----	u0	0

Рисунок 14. Выход из программы.

# Фрагменты листинга программы

Logic.cs

---

```
1 using System;
2
3 namespace kursovaya_emm
4 {
5     internal class Logic
6     {
7         double coef = 0.2;
8         double[] P = new double[6];
9         double[] zamena = new double[6];
10        double[] starting_value = new double[2];
11        double[,] P4_min_1 = new double[2, 4];
12        double[,] P3_min_1 = new double[2, 4];
13        double[,] P2_min_1 = new double[2, 4];
14        double[,] P4_min_2 = new double[2, 2];
15        double[,] P3_min_2 = new double[2, 2];
16        double[,] P2_min_2 = new double[2, 2];
17        double[] P1_min = new double[2];
18        private double getP(int i)
19        {
20            return starting_value[0] + starting_value[1] * (i - 1);
21        }
22        private double[,] getP4_min_1(double[,] P_min, double P_4, double P_5)
23        {
24            P_min[0, 0] = coef * P_4;
25            P_min[0, 1] = -1.0 * P_5;
26            P_min[1, 0] = P_4;
27            P_min[1, 1] = -1.0 * P_4;
28            P_min[1, 2] = coef * P_4;
29            P_min[1, 3] = -1.0 * P_5 / 2.0;
30            return P_min;
31        }
32        private double[,] getP_min_1(double[,] P_min, double P_4, double P_5, double
33        ↪ value)
34        {
35            P_min[0, 0] = coef * P_4;
36            P_min[0, 1] = -1.0 * P_5;
37            P_min[1, 0] = P_4;
38            P_min[1, 1] = -1.0 * P_4;
39            P_min[1, 2] = coef * P_4;
40            P_min[1, 3] = value;
41            return P_min;
42        }
43        private double[,] getP_min_2(double[,] P_min_1, double[,] P_min_2)
```

```

44     P_min_2[0, 0] = P_min_1[0, 0];
45     P_min_2[0, 1] = P_min_1[0, 1];
46     P_min_2[1, 0] = P_min_1[1, 0] + P_min_1[1, 2] + P_min_1[1, 3];
47     P_min_2[1, 1] = P_min_1[1, 1];
48     return P_min_2;
49 }
50 private double getMain_k_t_value_1(double[,] P_min, int k_t)
51 {
52     return P_min[0, 0] * (k_t + 1);
53 }
54 private double getMain_k_t_value_2(double[,] P_min, int k_t)
55 {
56     return P_min[0, 1] * Math.Pow(2, -1 * (k_t + 1));
57 }
58 private double getMain_k_t_value_3(double[,] P_min, int k_t)
59 {
60     return P_min[0, 0] * (k_t + 1) + P_min[0, 1] * Math.Pow(2, -1 * (k_t + 1));
61 }
62 private string getMain_k_t_sum_text(string value_1, string value_2)
63 {
64     return Convert.ToString(Convert.ToDouble(value_1) +
65         ↪ Convert.ToDouble(value_2));
66 }
67 private string getMain_k_t_subtract_text(string value_1, string value_2)
68 {
69     return Convert.ToString(Convert.ToDouble(value_1) -
70         ↪ Convert.ToDouble(value_2));
71 }
72 private string getMain_k_t_min_text(string value_1, string value_2)
73 {
74     if (Convert.ToDouble(value_1) - Convert.ToDouble(value_2) < 0)
75     {
76         return value_1;
77     }
78     else
79     {
80         return value_2;
81     }
82 }
83 private string getMain_k_t_compare_text(string value_1, string value_2)
84 {
85     if (Convert.ToDouble(value_1) == Convert.ToDouble(value_2))
86     {
87         return "0";
88     }
89     else
90     {
91         return "1";
92     }
93 }

```

```

90     }
91 }
92 private double getMain_k_t_value_5(double[,] P_min, int k_t)
93 {
94     return P_min[1, 1] * Math.Pow(2, -1 * (k_t));
95 }
96 private double getMain_k_t_value_6(double[,] P_min, int k_t)
97 {
98     return P_min[1, 0] + P_min[1, 1] * Math.Pow(2, -1 * (k_t));
99 }
100 private int getMain_k_t_value_7(double[,] P_min, int k_t)
101 {
102     if (getMain_k_t_value_3(P_min, k_t) == getMain_k_t_value_8(P_min, k_t))
103     {
104         return 0;
105     }
106     else
107     {
108         return 1;
109     }
110 }
111 private double getMain_k_t_value_8(double[,] P_min, int k_t)
112 {
113     return Math.Min(getMain_k_t_value_3(P_min, k_t), getMain_k_t_value_6(P_min,
114         ↪ k_t));
115 }
116 private string getR_k_last(double P, string t, double z)
117 {
118     return Convert.ToString(coef * P * (Convert.ToDouble(t) * z + 1));
119 }
120 //main table info
121 //u0
122 //first value
123 public static string main_k4_t1_u0_first_value_solve,
124     ↪ main_k4_t2_u0_first_value_solve, main_k4_t3_u0_first_value_solve,
125     ↪ main_k3_t1_u0_first_value_solve,
126     main_k3_t2_u0_first_value_solve, main_k2_t1_u0_first_value_solve,
127     ↪ main_k1_t0_u0_first_value_solve;
128 //second value
129 public static string main_k4_t1_u0_second_value_solve,
130     ↪ main_k4_t2_u0_second_value_solve, main_k4_t3_u0_second_value_solve,
131     ↪ main_k3_t1_u0_second_value_solve,
132     main_k3_t2_u0_second_value_solve, main_k2_t1_u0_second_value_solve,
133     ↪ main_k1_t0_u0_second_value_solve;
134 //u1
135 //first value
136 public static string main_k4_t1_u1_first_value_solve,
137     ↪ main_k4_t2_u1_first_value_solve, main_k4_t3_u1_first_value_solve,
138     ↪ main_k3_t1_u1_first_value_solve,

```

```

130         main_k3_t2_u1_first_value_solve, main_k2_t1_u1_first_value_solve,
           ↪ main_k1_t1_u1_first_value_solve;
131 //second value
132 public static string main_k4_t1_u1_second_value_solve,
           ↪ main_k4_t2_u1_second_value_solve, main_k4_t3_u1_second_value_solve,
           ↪ main_k3_t1_u1_second_value_solve,
133         main_k3_t2_u1_second_value_solve, main_k2_t1_u1_second_value_solve,
           ↪ main_k1_t0_u1_second_value_solve;
134 //second table info
135 public static string last_t_k3_solve, last_t_k4_solve, last_t_k5_solve;
136 public static string last_p_k2_solve, last_p_k3_solve, last_p_k4_solve;
137 public static string last_fi_k2_solve, last_fi_k3_solve, last_fi_k4_solve,
           ↪ last_fi_k5_solve;
138 public static string last_r_k1_solve, last_r_k2_solve, last_r_k3_solve,
           ↪ last_r_k4_solve;
139 public static string var_select_pk_first_value, var_select_pk_second_value;
140 public static bool aoutomat_solve = false, solve_was_started = false;
141 public static string automat_pass = "20112001"; //Password
142 public void SolveTheTask(double pk_first_value, double pk_second_value)
143 {
144     solve_was_started = true;
145     starting_value[0] = pk_first_value;
146     starting_value[1] = pk_second_value;
147     P[5] = getP(5);
148     P[4] = getP(4);
149     P[3] = getP(3);
150     P[2] = getP(2);
151     P[1] = getP(1);
152     //Заполнение P4_min
153     P4_min_1 = getP4_min_1(P4_min_1, P[4], P[5]);
154     P4_min_2 = getP_min_2(P4_min_1, P4_min_2);
155     //Заполнение k = 4
156     main_k4_t1_u0_first_value_solve =
           ↪ Convert.ToString(getMain_k_t_value_1(P4_min_2, 1));
157     main_k4_t2_u0_first_value_solve =
           ↪ Convert.ToString(getMain_k_t_value_1(P4_min_2, 2));
158     main_k4_t3_u0_first_value_solve =
           ↪ Convert.ToString(getMain_k_t_value_1(P4_min_2, 3));
159
160     main_k4_t1_u0_second_value_solve = Convert.ToString(-1 *
           ↪ getMain_k_t_value_2(P4_min_2, 1));
161     main_k4_t2_u0_second_value_solve = Convert.ToString(-1 *
           ↪ getMain_k_t_value_2(P4_min_2, 2));
162     main_k4_t3_u0_second_value_solve = Convert.ToString(-1 *
           ↪ getMain_k_t_value_2(P4_min_2, 3));
163
164     main_k4_t1_u1_first_value_solve = Convert.ToString(P4_min_2[1, 0]);
165     main_k4_t2_u1_first_value_solve = Convert.ToString(P4_min_2[1, 0]);

```

```

166     main_k4_t3_u1_first_value_solve = Convert.ToString(P4_min_2[1, 0]);
167
168     main_k4_t1_u1_second_value_solve = Convert.ToString(-1 *
169     ↪   getMain_k_t_value_5(P4_min_2, 1));
170     main_k4_t2_u1_second_value_solve = Convert.ToString(-1 *
171     ↪   getMain_k_t_value_5(P4_min_2, 2));
172     main_k4_t3_u1_second_value_solve = Convert.ToString(-1 *
173     ↪   getMain_k_t_value_5(P4_min_2, 3));
174
175     string main_p_k4_t1_solve_h = Convert.ToString(getMain_k_t_value_8(P4_min_2,
176     ↪   1));
177     string main_p_k4_t2_solve_h = Convert.ToString(getMain_k_t_value_8(P4_min_2,
178     ↪   2));
179     string main_p_k4_t3_solve_h = Convert.ToString(getMain_k_t_value_8(P4_min_2,
180     ↪   3));
181
182     string main_u_k4_t1_solve_h = Convert.ToString(getMain_k_t_value_7(P4_min_2,
183     ↪   1));
184     string main_u_k4_t2_solve_h = Convert.ToString(getMain_k_t_value_7(P4_min_2,
185     ↪   2));
186     string main_u_k4_t3_solve_h = Convert.ToString(getMain_k_t_value_7(P4_min_2,
187     ↪   3));
188
189     //Заполнение P3_min
190     P3_min_1 = getP_min_1(P3_min_1, P[3], P[4],
191     ↪   Convert.ToDouble(main_p_k4_t1_solve_h));
192     P3_min_2 = getP_min_2(P3_min_1, P3_min_2);
193
194     //Заполнение P3
195
196     main_k3_t1_u0_first_value_solve =
197     ↪   Convert.ToString(getMain_k_t_value_1(P3_min_2, 1));
198     main_k3_t2_u0_first_value_solve =
199     ↪   Convert.ToString(getMain_k_t_value_1(P3_min_2, 2));
200
201     main_k3_t1_u0_second_value_solve = main_p_k4_t2_solve_h;
202     main_k3_t2_u0_second_value_solve = main_p_k4_t3_solve_h;
203
204     string main_k3_t1_u0_result_solve_h =
205     ↪   getMain_k_t_sum_text(main_k3_t1_u0_first_value_solve,
206     ↪   main_k3_t1_u0_second_value_solve);
207     string main_k3_t2_u0_result_solve_h =
208     ↪   getMain_k_t_sum_text(main_k3_t2_u0_first_value_solve,
209     ↪   main_k3_t2_u0_second_value_solve);
210
211     main_k3_t1_u1_first_value_solve = Convert.ToString(P3_min_2[1, 0]);
212     main_k3_t2_u1_first_value_solve = Convert.ToString(P3_min_2[1, 0]);
213

```



```

198     main_k3_t1_u1_second_value_solve = Convert.ToString(-1 *
        ↳ getMain_k_t_value_5(P3_min_2, 1));
199     main_k3_t2_u1_second_value_solve = Convert.ToString(-1 *
        ↳ getMain_k_t_value_5(P3_min_2, 2));
200
201     string main_k3_t1_u1_result_solve_h =
        ↳ getMain_k_t_subtract_text(main_k3_t1_u1_first_value_solve,
        ↳ main_k3_t1_u1_second_value_solve);
202     string main_k3_t2_u1_result_solve_h =
        ↳ getMain_k_t_subtract_text(main_k3_t2_u1_first_value_solve,
        ↳ main_k3_t2_u1_second_value_solve);
203
204     string main_p_k3_t1_solve_h =
        ↳ getMain_k_t_min_text(main_k3_t1_u0_result_solve_h,
        ↳ main_k3_t1_u1_result_solve_h);
205     string main_p_k3_t2_solve_h =
        ↳ getMain_k_t_min_text(main_k3_t2_u0_result_solve_h,
        ↳ main_k3_t2_u1_result_solve_h);
206
207     string main_u_k3_t1_solve_h =
        ↳ getMain_k_t_compare_text(main_k3_t1_u0_result_solve_h,
        ↳ main_p_k3_t1_solve_h);
208     string main_u_k3_t2_solve_h =
        ↳ getMain_k_t_compare_text(main_k3_t2_u0_result_solve_h,
        ↳ main_p_k3_t2_solve_h);
209
210     //Заполнение P2_min
211     P2_min_1 = getP_min_1(P2_min_1, P[2], P[3],
        ↳ Convert.ToDouble(main_p_k3_t1_solve_h));
212     P3_min_2 = getP_min_2(P2_min_1, P2_min_2);
213
214     //Заполнение P2
215
216     main_k2_t1_u0_first_value_solve =
        ↳ Convert.ToString(getMain_k_t_value_1(P2_min_2, 1));
217
218     main_k2_t1_u0_second_value_solve = main_p_k3_t2_solve_h;
219
220     string main_k2_t1_u0_result_solve_h =
        ↳ getMain_k_t_sum_text(main_k2_t1_u0_first_value_solve,
        ↳ main_k2_t1_u0_second_value_solve);
221
222     main_k2_t1_u1_first_value_solve = Convert.ToString(P2_min_2[1, 0]);
223
224
225     main_k2_t1_u1_second_value_solve = Convert.ToString(-1 *
        ↳ getMain_k_t_value_5(P2_min_2, 1));
226

```

```

227
228     string main_k2_t1_u1_result_solve_h =
        ↳ getMain_k_t_subtract_text(main_k2_t1_u1_first_value_solve,
        ↳ main_k2_t1_u1_second_value_solve);
229
230     string main_p_k2_t1_solve_h =
        ↳ getMain_k_t_min_text(main_k2_t1_u0_result_solve_h,
        ↳ main_k2_t1_u1_result_solve_h);
231
232     string main_u_k2_t1_solve_h =
        ↳ getMain_k_t_compare_text(main_k2_t1_u0_result_solve_h,
        ↳ main_p_k2_t1_solve_h);
233
234
235
236     //Заполнение P1_min
237     P1_min[0] = coef * P[1];
238     P1_min[1] =
        ↳ Convert.ToDouble(getMain_k_t_sum_text(main_k2_t1_u0_first_value_solve,
        ↳ main_k2_t1_u0_second_value_solve));
239
240     //Заполнение P1
241
242     main_k1_t0_u0_first_value_solve = Convert.ToString(P1_min[0]);
243
244     if (Convert.ToDouble(main_k2_t1_u0_result_solve_h) >
        ↳ Convert.ToDouble(main_k2_t1_u1_result_solve_h))
245     {
246         main_k1_t0_u0_second_value_solve = main_k2_t1_u1_result_solve_h;
247     }
248     else
249     {
250         main_k1_t0_u0_second_value_solve = main_k2_t1_u0_result_solve_h;
251     }
252
253
254     string main_k1_t0_u0_result_solve_h =
        ↳ getMain_k_t_sum_text(main_k1_t0_u0_first_value_solve,
        ↳ main_k1_t0_u0_second_value_solve);
255 }
256 }

```

---

main\_window.cs

---

```

1 using System;
2 using System.Windows.Forms;
3 using System.Diagnostics;

```

4

```

5 namespace kursovaya_emm
6 {
7     public partial class main_window : Form
8     {
9         public main_window()
10        {
11            InitializeComponent();
12        }
13        Logic logic = new Logic();
14        private void start_check_Click(object sender, EventArgs e)
15        {
16            if (Logic.aoutomat_solve)
17            {
18                color_form_white();
19                if (pk_first_value.Text == "?" || pk_second_value.Text == "?")
20                {
21                    if (pk_first_value.Text == "?")
22                        pk_first_value.BackColor = System.Drawing.Color.OrangeRed;
23                    if (pk_second_value.Text == "?")
24                        pk_second_value.BackColor = System.Drawing.Color.OrangeRed;
25                }
26                else
27                {
28
29                    logic.SolveTheTask(Convert.ToDouble(pk_first_value.Text),
30                        ↪ Convert.ToDouble(pk_second_value.Text));
31
32                    //Заполнение k = 4
33                    secondTable secondT = new secondTable();
34                    secondT.ShowDialog();
35                }
36            }
37            else
38            {
39                color_form_white();
40                if (pk_first_value.Text == "?" || pk_second_value.Text == "?")
41                {
42                    if (pk_first_value.Text == "?")
43                        pk_first_value.BackColor = System.Drawing.Color.OrangeRed;
44                    if (pk_second_value.Text == "?")
45                        pk_second_value.BackColor = System.Drawing.Color.OrangeRed;
46                }
47                else
48                {
49                    int count_errors = 0;
50                    Logic logic = new Logic();
51                    logic.SolveTheTask(Convert.ToDouble(pk_first_value.Text),
52                        ↪ Convert.ToDouble(pk_second_value.Text));

```

```

51      //Заполнение k = 4
52      if (main_k4_t1_u0_first_value.Text !=
53          ↪ Logic.main_k4_t1_u0_first_value_solve)
54      {
55          main_k4_t1_u0_first_value.BackColor =
56          ↪ System.Drawing.Color.OrangeRed;
57          count_errors++;
58      }
59      ...
60      //Заполнение k =3
61      if (main_k3_t1_u0_first_value.Text !=
62          ↪ Logic.main_k3_t1_u0_first_value_solve)
63      {
64          main_k3_t1_u0_first_value.BackColor =
65          ↪ System.Drawing.Color.OrangeRed;
66          count_errors++;
67      }
68      ...
69      //Заполнение k =2
70      if (main_k2_t1_u0_first_value.Text !=
71          ↪ Logic.main_k2_t1_u0_first_value_solve)
72      {
73          main_k2_t1_u0_first_value.BackColor =
74          ↪ System.Drawing.Color.OrangeRed;
75          count_errors++;
76      }
77      ...
78      //Заполнение k =1
79      if (main_k1_t0_u0_first_value.Text !=
80          ↪ Logic.main_k1_t0_u0_first_value_solve)
81      {
82          main_k1_t0_u0_first_value.BackColor =
83          ↪ System.Drawing.Color.OrangeRed;
84          count_errors++;
85      }
86      ...
87      if (count_errors > 0)
88      {
89          MessageBox.Show("Неправильное заполнение таблицы! Найдено ошибок:
90          ↪ " + Convert.ToString(count_errors));
91      }
92      else
93      {
94          secondTable secondT = new secondTable();
95          secondT.ShowDialog();
96      }
97      }
98      }
99  }

```

```

90     }
91     //Валидация полей
92
93     private void first_p_Validated(object sender, EventArgs e)
94     {
95         color_form_white();
96         try
97         {
98             double firrst_val = Convert.ToDouble((sender as TextBox).Text);
99             if(pk_second_value.Text != "?")
100             {
101                 logic.SolveTheTask(firrst_val,
102                     ↪ Convert.ToDouble(pk_second_value.Text));
103                 check_table();
104             }
105         }
106         catch (Exception)
107         {
108             Logic.solve_was_started = false;
109             (sender as TextBox).Text = "?";
110         }
111     }
112     private void second_p_Validated(object sender, EventArgs e)
113     {
114         color_form_white();
115         try
116         {
117             double second_val = Convert.ToDouble((sender as TextBox).Text);
118             if (pk_first_value.Text != "?")
119             {
120                 logic.SolveTheTask(Convert.ToDouble(pk_first_value.Text),
121                     ↪ second_val);
122                 check_table();
123             }
124         }
125         catch (Exception)
126         {
127             Logic.solve_was_started = false;
128             (sender as TextBox).Text = "?";
129         }
130     }
131     private void text_boxs_validate(object sender)
132     {
133         try
134         {
135             Convert.ToDouble((sender as TextBox).Text);
136         }
137         catch (Exception)

```

```

136         {
137             (sender as TextBox).Text = "?";
138         }
139     }
140     private void text_box_KeyPress(object sender, KeyPressEventArgs e)
141     {
142         if (e.KeyChar == '.') e.KeyChar = ',';
143     }
144 }
145 private void text_boxs_MouseClick_Enter(object sender, EventArgs e)
146 {
147     (sender as TextBox).BackColor = System.Drawing.Color.White;
148     if ((sender as TextBox).Text == "?")
149         (sender as TextBox).Text = "";
150 }
151 private void text_boxs_MouseClick(object sender, MouseEventArgs e)
152 {
153     (sender as TextBox).BackColor = System.Drawing.Color.White;
154     if ((sender as TextBox).Text == "?")
155         (sender as TextBox).Text = "";
156 }
157 private void menu_out_button_Click(object sender, EventArgs e)
158 {
159     this.Close();
160 }
161 private void menu_spravka_Click(object sender, EventArgs e)
162 {
163     try
164     {
165         Process.Start("help.pdf");
166     }
167     catch (Exception)
168     {
169
170         MessageBox.Show("Справка по программе не найдена!");
171     }
172 }
173 }
174 private void menu_work_mod_handle_mod_enter_data_Click(object sender, EventArgs
175 ↵ e)
176 {
177     clear_form();
178     color_form_white();
179 }
180 private void menu_work_mod_handle_mod_from_file_Click(object sender, EventArgs e)
181 {
182     VarSelect varSelect = new VarSelect();
183     varSelect.ShowDialog();

```

```

183         if (Logic.var_select_pk_first_value != "-" ||
    ↪ Logic.var_select_pk_second_value != "-")
184     {
185         clear_form();
186         color_form_white();
187         pk_first_value.Text = Logic.var_select_pk_first_value;
188         pk_second_value.Text = Logic.var_select_pk_second_value;
189     }
190 }
191 private void menu_work_mod_auto_Click(object sender, EventArgs e)
192 {
193     if (Logic.aoutomat_solve)
194     {
195         Logic.aoutomat_solve = false;
196         menu_work_mod_auto.Text = "Рассчитать автоматически (off)";
197     }
198     else
199     {
200         PassCheck passCheck = new PassCheck();
201         passCheck.ShowDialog();
202         if (Logic.aoutomat_solve)
203         {
204             menu_work_mod_auto.Text = "Рассчитать автоматически (on)";
205         }
206     }
207 public void check_table()
208 {
209     if (main_k4_t1_u0_first_value.Text != Logic.main_k4_t1_u0_first_value_solve
    ↪ && main_k4_t1_u0_first_value.Text != "?" )
210     {
211         main_k4_t1_u0_first_value.BackColor = System.Drawing.Color.OrangeRed;
212     }
213 }
214 ...
215 //Заполнение k =3
216 if (main_k3_t1_u0_first_value.Text != Logic.main_k3_t1_u0_first_value_solve
    ↪ && main_k3_t1_u0_first_value.Text != "?")
217 {
218     main_k3_t1_u0_first_value.BackColor = System.Drawing.Color.OrangeRed;
219 }
220 }
221 ...
222 //Заполнение k =2
223 if (main_k2_t1_u0_first_value.Text != Logic.main_k2_t1_u0_first_value_solve
    ↪ && main_k2_t1_u0_first_value.Text != "?")
224 {
225     main_k2_t1_u0_first_value.BackColor = System.Drawing.Color.OrangeRed;
226

```

```
227         }
228         ...
229         //Заполнение k =1
230         if (main_k1_t0_u0_first_value.Text != Logic.main_k1_t0_u0_first_value_solve
231             ⇨ && main_k1_t0_u0_first_value.Text != "?")
232         {
233             main_k1_t0_u0_first_value.BackColor = System.Drawing.Color.OrangeRed;
234         }
235         ...
236     }
237 }
```

---



## Заключение

В ходе курсовой работы были изучены материалы по задаче замены оборудования и было написано графическое приложение для нахождения плана эксплуатации оборудования, при котором затраты за весь период минимальны.

# Литература

- [1] Львовский, С.М. Набор и верстка в системе  $\text{\LaTeX}$  [Текст] / С.М. Львовский. М.: МЦНМО, 2006. — 448 с.
- [2] Лунгу К. Н. Линейное программирование. Руководство к решению задач. /Лунгу К. Н. - М.: ФИЗМАТЛИТ, 2005. - 128 с.