

**УДК 004.588**

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ ЕЛЕМЕНТІВ  
ТРЕНАЖЕРА З ПОБУДОВИ МАТЕМАТИЧНОЇ МОДЕЛІ  
КОМБІНАТОРНОЇ ОПТИМІЗАЦІЙНОЇ  
“ЗАДАЧІ ДИРЕКТОРА”**

**В. В. Куркін**, студент гр. І-41, спеціальності «Інформатика»

Вищий навчальний заклад Укоопспілки «Полтавський  
університет економіки і торгівлі»  
caucho228@gmail.com

**О. О. Ємець**, д.ф.-м.н., професор

Вищий навчальний заклад Укоопспілки «Полтавський  
університет економіки і торгівлі»  
yemetsli@ukr.net

*Запропоновано алгоритм і створено тренажер на  
тему «Задача директора» для дистанційного курсу «Методи  
оптимізації та дослідження операцій».*

*Kyrkin V. V., Iemets O.O. ALGORITHMIZATION AND  
PROGRAMMING OF ELEMENTS OF TRAINER OF  
CONSTRUCTION OF MATHEMATICAL MODEL OF  
COMBINATOR OPTIMIZATION “DIRECTOR’S PROBLEM”. An  
algorithm is proposed and a trainer is created on the topic  
"Director’s problem" for the distance course "Methods of  
optimization and operations research".*

*Ключові слова:* АЛГОРИТМ, ТРЕНАЖЕР, ЗАДАЧА  
ДИРЕКТОРА, ПРОГРАМУВАННЯ.

*Keywords:* ALGORITM, TRAINER, PROGRAMMING,  
DIRECTOR'S PROBLEM.

В доповіді викладена постановка задачі і частина алгоритму роботи тренажера. Метою роботи є розробка алгоритму та програмного забезпечення тренажера з теми «Задача директора» дистанційного навчального курсу «Методи оптимізації та

дослідження операцій». Тренажер створюється для вивчення цієї теми. При ознайомленні з темою «Задача директора» були використані лекції з дисципліни «Методи оптимізації та дослідження операцій»[1]. Як в дистанційному курсі з «Методів оптимізації та дослідження операцій» ПУЕТ, так і в Інтернеті (в україномовному сегменті) немає тренажера на тему «Задача Директора». Тому створення такого тренажера є актуальним. В тренажері повинно імітуватися процес розв'язування задачі. Алгоритм представлено далі.

Алгоритм тренажера.

Крок 1. На екрані умова задачі (у подальшому умова має буди завжди на екрані до завершення складання моделі). В приймальні  $k$  осіб, час прийому яких відомий:  $T_1, \dots, T_k$  відповідно. Директору треба так організувати прийом, щоб загальний час очікування був мінімальний.

Далі – запитання:

Що необхідно визначити в результаті задачі?

- а)  $X_j$  час, що йде на прийом відвідувача, що зайшов останнім;
- б)  $X_j$  час, що йде на прийом відвідувача, що зайшов першим;
- в)  $X_j$  час, що йде на прийом відвідувача, що зайшов  $j$ -им;
- г) час, що йде на прийом  $k$  осіб;

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь. ( $X_j$  час, що йде на прийом відвідувача, що зайшов  $j$ -им;)

Інакше перехід до наступного кроку.

Крок 2-й. З'являється наступне питання: “Якщо зайшов перший відвідувач, скільки залишилося в приймальні?”

- а) 1;
- б) 2;
- в)  $k-1$ ;
- г)  $k+1$ .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь:  $k-1$ .

Інакше перехід до наступного кроку.

Крок 3-й. З'являється наступне питання: “ Скільки часу чекає кожен відвідувач, коли зайшов перший?”

- а)  $T_1$ ;
- б)  $T_j$ ;
- в)  $X_j$ ;
- г)  $X_1$ .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь:  $X_1$ .

Інакше перехід до наступного кроку.

Крок 4-й. З'являється наступне питання: “ Скільки часу чекають усі відвідувачі, коли зайшов перший відвідувач?”

- а)  $(k-1)*T_1$ ;
- б)  $(k-1)*X_1$ ;
- в)  $X_1$ ;
- г)  $T_1$ .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь:  $(k-1)*X_1$ .

Інакше перехід до наступного кроку.

Крок 5.

З'являється наступне питання: “ Скільки часу чекає кожен відвідувач, коли приймають  $j$ -го відвідувача?”

- а)  $X_j$ ;
- б)  $X_1$ ;
- в)  $T_1$ ;
- г)  $T_j$ .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь:  $X_j$ .

Інакше перехід до наступного кроку.

Крок 6.

З'являється наступне питання: “ Скільки часу чекають усі відвідувачі, коли зайшов  $j$ -ий відвідувач?”

- а)  $(k-1)*X_1$ ;
- б)  $(k-j)*X_j$ ;
- в)  $(k-j)*X_1$ ;
- г)  $(k-j)*T_1$ .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь:  $(k-j)*X_j$

Інакше перехід до наступного кроку.

Крок 7.

З'являється наступне завдання: “Який вигляд має цільова функція? (Внесіть в активні комірки значення цільової функції)”

Потрібно в порожні комірки вписати певне значення.

$(\quad)*X_1+(\quad)*X_2+\dots+(\quad)*X_j+\dots+(\quad)*X_{k-1}+(\quad)*X_k$

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь:  $(k-1)*X_1+(k-2)*X_2+\dots+(k-j)*X_j+\dots+(1)*X_{k-1}+(0)*X_k$ .

Інакше перехід до наступного кроку.

Крок 8.

З'являється наступне питання: “Цільова функція буде мінімізуватися чи максимізуватися?”

а)Мінімізуватися;

б)Максимізуватися.

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: мінімізуватися.

Інакше – перехід до наступного кроку.

Крок 9.

З'являється таке питання: “Яка з осіб може прийматися першою?”

а)Та у якої час обслуговування  $T_1$ ;

б)Будь-хто;

в)Та у якої час прийому найменший.

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: будь-хто.

Інакше перехід до наступного кроку.

Крок 10.

З'являється наступне питання: “Яка з осіб може прийматися наступною?”

а)Будь-хто;

б) Будь-хто з тих, що залишилися;

в) Та особа у якої час прийому  $T_2$ .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: будь-хто з тих, що залишилися.

Інакше перехід до наступного кроку.

Крок 11.

З'являється наступне питання: “Яка з осіб може прийматися  $j$ -ою?”

а) Та особа, у якої час прийому  $T_j$ ;

б) Будь-хто;

в) Будь-хто з тих, що залишилися.

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: будь-хто з тих, що залишилися.

Інакше перехід до наступного кроку.

Крок 12.

З'являється наступне питання: “Отже, з якої комбінаторної множини вибирається вектор невідомих  $x=(X_1, \dots, X_k)$ ?”

а) Множина перестановок;

б) Множина розміщень, де  $T=\{T_1, \dots, T_k\}$ , а  $v$ -кількість різних елементів в  $T$ .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: множина перестановок.

Інакше перехід до наступного кроку.

Крок 13.

З'являється наступне питання: “Чи є інші обмеження в задачі директора?”

а) Ні;

б) Так.

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: ні.

Інакше з'являється вікно з повідомленням про завершення роботи тренажера.

Розроблений алгоритм реалізовано у вигляді програми на

мові С# у програмному середовищі Visual Studio 2017 Community, яка містить близько 1000 рядків програмного коду.

Розроблено новий тренажер по створенню математичної моделі «Задачі директора» для дистанційного курсу «Методи операцій та дослідження операцій». Цей тренажер можна використовувати як в дистанційному так і в стаціонарному навчанні студента.

### **Список використаних джерел**

1. Ємець О.О. Методи оптимізації та дослідження операцій [Електронний ресурс] : навчально-методичний посібник за кредитно-модульною організацією навчального процесу/ О. О. Ємець, Т. О. Парфьонова. – Полтава : ПУЕТ, 2013. Режим доступу: [http://elib.puet.edu.ua/action.php?kt\\_path\\_info=lm.web.view&fDocumentId=670571](http://elib.puet.edu.ua/action.php?kt_path_info=lm.web.view&fDocumentId=670571)
2. Ємець О.О. Елементи комбінаторної оптимізації: Навчально-методичний посібник / О.О. Ємець. – Полтава: РВВ ПУСКУ, 2009. – 23 с.
3. Рихтер Дж. Язык программирования С#. Классика Computers Science/ Дж. Рихтер. – Лейкерузен: Addison Wesley, 2011. – 715 с.
4. Хейлсберг А. Изучаем С#/ А. Хейлсберг. – Нью-Йорк: Питер, 2013. – 537 с.