

DOI: 10.36910/6775-2524-0560-2020-40-11

УДК: 004.421

Пахальчук Євгеній Вікторович, асистент

<https://orcid.org/0000-0001-9902-0738>

Приазовський державний технічний університет, м. Маріуполь, Україна

## СИСТЕМА АДАПТИВНОГО ПОШУКУ ОПТИМАЛЬНОГО МАРШРУТУ КОМІРНИКА ДЛЯ ВИДАЧІ ТОВАРІВ ЗА НАКЛАДНОЮ

**Пахальчук Е. В. Система адаптивного поиска оптимального маршрута кладовщика для выдачи товаров по накладной.** В данной работе описана система адаптивного поиска оптимального пути, направленная на упрощение ориентирования кладовщика на складе в момент сбора товаров по накладной. Рассмотрены функциональные возможности системы адаптивного поиска оптимального маршрута для кладовщика. Проанализирован набор алгоритмов поиска кратчайшего пути. Приведена условная реализация системы на основе алгоритма поиска кратчайшего пути  $A^*$ .

**Ключевые слова:** поиск оптимального маршрута, алгоритм поиска  $A^*$ , адаптивная системы поиска, мобильный подход.

**Пахальчук Є. В. Система адаптивного пошуку оптимального маршруту комірника для видачі товарів за накладною.** У даній роботі описана система адаптивного пошуку оптимального шляху, спрямована на спрощення орієнтування комірника на складі в момент збору товарів за накладною. Розглянуто функціональні можливості системи адаптивного пошуку оптимального маршруту для комірника. Проаналізовано набір алгоритмів пошуку найкоротшого шляху. Наведена умовна реалізація системи на основі алгоритму пошуку найкоротшого шляху  $A^*$ .

**Ключові слова:** пошук оптимального маршруту, алгоритм пошуку  $A^*$ , адаптивна системи пошуку, мобільний підхід.

**Pahalchuk E. System of adaptive search for the optimal storerer's route for dispensing goods on the way.** This paper describes an adaptive search system for the optimal path, aimed at simplifying the orientation of the storekeeper in the ware house at the time of collecting goods on the invoice. The functional capabilities of the system of adaptive search for the optimal route for the storekeeper are considered. A set of algorithms for finding the shortest path is analyzed. A conditional implementation of the system based on the shortest path search algorithm  $A^*$  is given.

**Keywords:** optimal route search,  $A^*$  search algorithm, adaptive search systems, mobile approach.

**Постановка наукової проблеми.** Сучасні складські системи в своїй роботі все частіше використовують системи управління складами WMS [1], або альтернативні системи зі схожим функціоналом. Суть роботи подібних систем полягає в забезпеченні автоматизації управління бізнес-процесами складської роботи. Вони зберігають у собі інформацію про товар, користувачів, накладних та інших документах, і обробляють її згідно з різними запитами. Все це зроблено для прискорення роботи комірника на таких складах. Одним з обов'язків комірника є видача товару згідно з накладною. І основне завдання, яке виникає у комірника в момент збору товарів зі списку, це оптимізація шляху переміщення по складу для мінімізації витраченого часу на збір. Чим менше часу буде витрачено на виконання однієї накладної, тим більше користувачів зможе обслужити комірник. У разі, якщо склад має невеликі розміри, а комірник вже давно на ньому працює, то пошук оптимального маршруту він здатний скласти самостійно. Але якщо розмір складу досить великий, і є висока різновид позицій, то самостійний прорахунок оптимального шляху збору товарів стає досить проблематичним. А якщо комірник тільки почав роботу на новому складі, то самостійне формування оптимального шляху практично неможливо. Також процес побудова оптимального маршруту ускладнюється переміщенням товарів всередині складу, коли товар міняє своє місце розташування на складі. Тому завдання з розробкою системи для побудови оптимального маршруту збору товарів з накладної є досить актуальним. Особливо для складів з великою площею, великий різновидом позицій і частими внутрішніми переміщеннями товару.

**Аналіз досліджень.** Завдання з пошуком оптимального маршруту для збору товарів за накладною можна вважати окремим випадком завдання комівояжера (traveling salesman problem) [2]. Завдання комівояжера отримало свою назву завдяки наступній ситуації. Комівояжеру потрібно відвідати певне число міст тільки один раз і повернутися назад в рідне місто на місце старту. Він знає відстані між кожною парою міст і хоче визначити такий порядок їх відвідування, щоб загальна пройдена відстань була мінімальною. Очевидно, що ситуація з комівояжером має кілька точок дотику із завданням збору товарів з накладною. Комірник починає своє пересування по складу з місця отримання накладної, його робочого місця. У накладній вказується список необхідних позицій, які йому треба видати. Після цього він починає свій рух по складу, відвідуючи ті ряди і полки, на яких

розташовується товар. Даний підхід дуже ефективний, так як практично повністю дозволяє вирішити поставлену задачу. Метод є неадаптивним і не здатний враховувати непередбачені зупини.

Так само при вирішенні завдань для пошуку найкоротшого шляху часто використовують алгоритм Дейкстри [3] і алгоритм A\* [4]. Алгоритм Дейкстри (також званий пошуком з рівномірною вартістю) дозволяє нам ставити пріоритети дослідження шляхів. Замість рівномірного дослідження всіх можливих шляхів він віддає перевагу шляхам з низькою вартістю. Ми можемо поставити зменшені витрати, щоб алгоритм рухався по дорогах, підвищену вартість, щоб уникати лісів і ворогів, і багато іншого. Коли вартість руху може бути різною, ми використовуємо його замість пошуку в ширину.

A\* - це модифікація алгоритму Дейкстри, оптимізована для єдиною кінцевої точки. Алгоритм Дейкстри може знаходити шляхи на всі точки, A\* знаходить шлях до однієї точки. Він віддає пріоритет шляхам, які ведуть ближче до мети. На рисунку 1 представлено порівняння роботи двох алгоритмів.

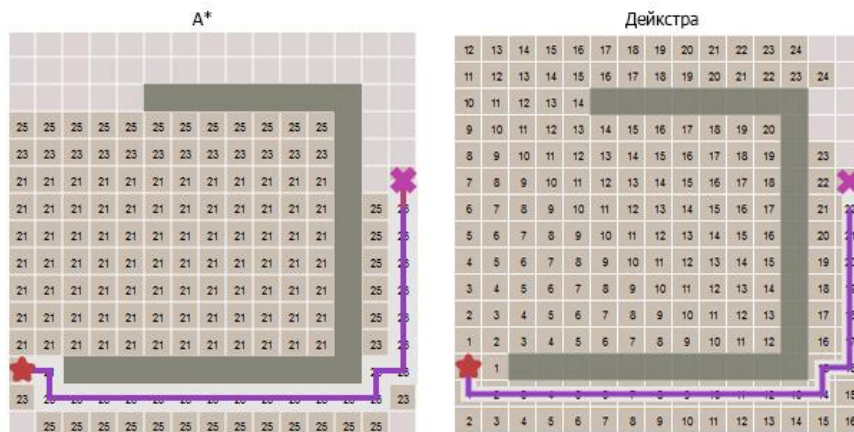


Рисунок 1 - Порівняння алгоритмів пошуку

Об'єктивно видно, що алгоритм A\* працює швидше ніж алгоритм Дейкстри. Це відбувається через те, що алгоритм Дейкстри обчислює відстань від початкової точки, а A\* використовує суму відстаней початкової і кінцевої точки. Так само A\* досліджує менше неактуальних областей пересування, що значно економить ресурси системи.

**Мета роботи.** Метою даної роботи є проектування та розробка модуля адаптивного пошуку оптимального маршруту комірника по складу, на основі оптимального алгоритму пошуку шляху A\*.

**Виклад основного матеріалу й обґрунтування отриманих результатів дослідження**

Перед рішенням завдання необхідно провести огляд існуючого обладнання. Так як подальша розробка системи і алгоритмів взаємодії між модулями може істотно відрізнитися в залежності від обраної платформи. Найпопулярнішими пристроями при роботі з товарами на складі є термінали збору даних [5]. Вони бувають різних типів і комплектацій, від пристроїв схожих на мобільні телефони, до портативного комп'ютера з клавіатурою і можливістю друку. Найбільш придатними варіантами для даного завдання виявилися пристрої на базі операційної системи Android з версією 7.1 [6]. Це дозволяє виробляти розробку мобільних додатків з використанням сучасних бібліотек та функцій. Подальша інформація в статті представлена з урахуванням розробки мобільного застосування для пошуку оптимального шляху.

Перед рішенням поставленого завдання, а саме адаптивного пошуку оптимального маршруту комірника по складу для видачі товарів за накладною, спочатку необхідно вирішити ряд дрібніших завдань:

- формування плану приміщення;
- налаштування плану складу шляхом присвоєння осередкам їх координат розташування;

Завдання формування і завантаження плану приміщення в систему детально буде розглянута в наступній роботі. Він полягає у автоматичному формуванні плану приміщення шляхом аналізу завантаженого зображення за допомогою бібліотеки OpenCV [7]. Зараз за основу буде взято типовий план складського приміщення, який представлений на рисунку 2. На рисунку видно кілька зон розмежувань:

- помаранчева зона - місце роботи комірника. Є початковою точкою побудови маршрутів пересування в момент отримання накладної.

– зелена зона - розташування стелажів на плані складу. Вона необхідна для того, щоб сформувати природні перешкоди в момент побудови маршруту. Надалі буде проводитися їх налаштування для розмітки осередків.

– блакитна зона - шлях по якому комірник може вільно переміщатися. Використовується системою для прокладки маршруту від точки А до точки Б.

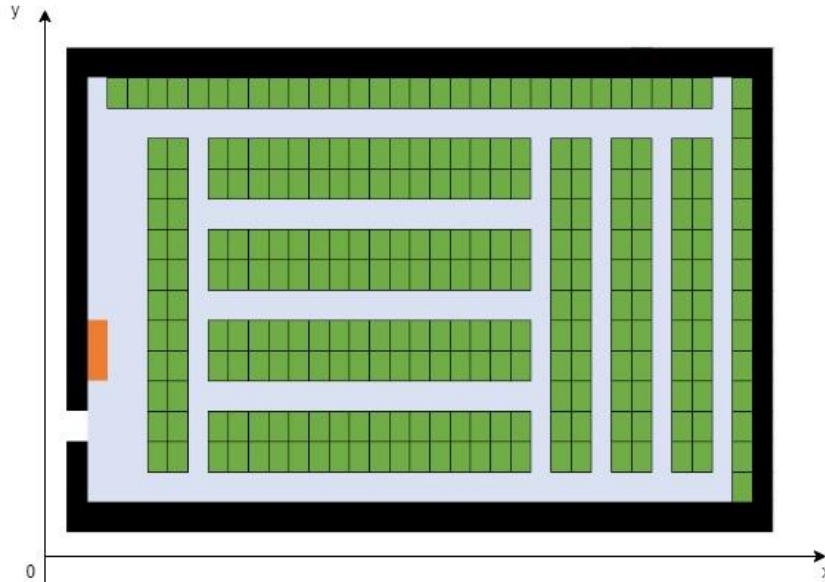


Рисунок 2 - Типовий план складу з розміткою

Після того, як карта складу була сформована і розділена на сегменти, можна приступати до її налаштування і адаптації під реальний склад. Для цього необхідно вказати місце розташування кожного осередку на стелажах. Щоб це реалізувати, карту складу необхідно відобразити на координатній площині. Для зручності будемо вважати нижній лівий кут карти як позиція  $(0,0)$  і подальшу розмітку виробляти відносно 0.

Присвоєння координатної позиції для осередку може бути сформовано двома способами:

- привласнення координат цільового осередку;
- привласнення координат групі осередків.

Для присвоєння координат цільового осередку необхідно обрати стелаж і номер комірки на ньому, після цього на плані складу необхідно натиснути на місце розташування самого осередку. Після цих дій в систему буде занесена інформація про фізичне розташування осередку на складі.

Для присвоєння координат групі осередків необхідно відкрити карту складу і вибрати позицію матеріального становища. Після того як позиція була відзначена, необхідно додати список осередків, які відповідають обраній позиції. У даній ситуації списку осередків задається однакове фізичне місце розташування. Це обумовлено тим, що на одній координаті в двовимірному просторі може перебувати кілька осередків один над одним. І якщо уявити стелаж і осередок в тривимірному просторі, то у осередку з'явилася б з координата, яка відповідає за відстань від підлоги. Але в даній задачі координата z не розглядається, тому що вона не впливає при розрахунку відстані між точками. Після налаштування карти складу система готова до експлуатації.

#### **Загальна структура системи адаптивного пошуку оптимального маршруту для комірника**

Розроблена система в цілому складається з групи модулів. Кожен з модулів відповідає за свою частину роботи:

- модуль «Сервер» зберігає накладні і товари, формує набори даних по фільтрам;
- модуль «Клієнт» містить в собі інструменти візуалізації, формування плану складу;
- модуль «Пошуку маршруту» виробляє аналіз товарів зазначених в накладній, прораховує оптимальні маршрути від стартової позиції для подальшої побудови маршруту.

На рисунку 3 представлена загальна схема структури системи, в якій відображений модуль адаптивного пошуку оптимального шляху для комірника.

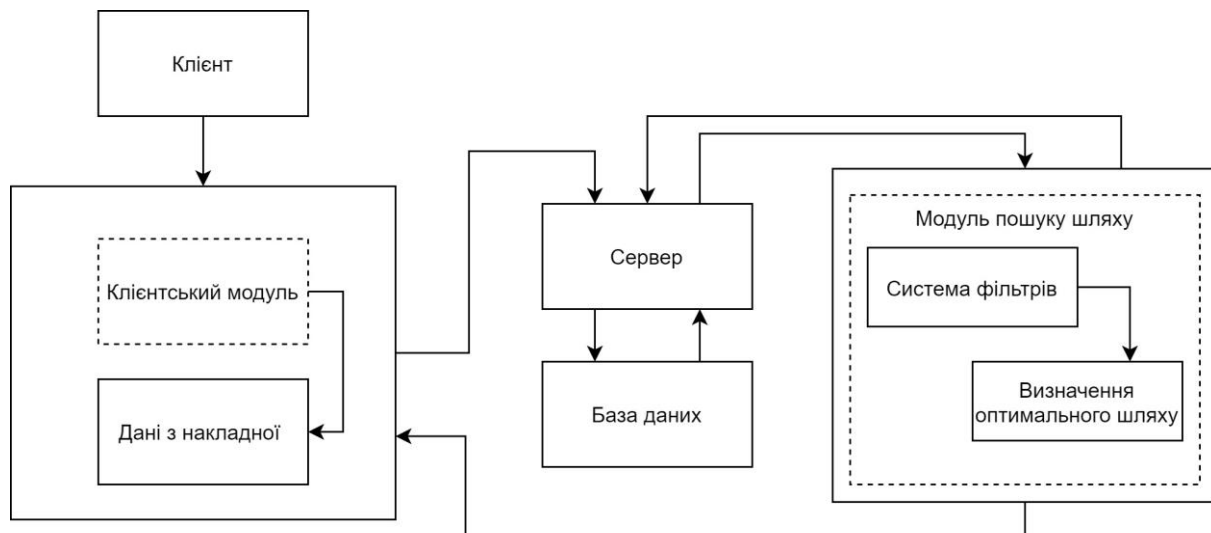


Рисунок 3 - Структурна схема системи адаптивного пошуку оптимального шляху

Схема складається зі стандартної клієнт-серверної архітектури з додатковим модулем адаптивного пошуку оптимального шляху. Робота клієнтської частини програми заснована на структурованому виведенні інформації користувачеві в момент роботи, візуалізації шляхів переміщення, віддаленому обліку товарів та інших функціях необхідних комерсанти. Так як пристрій, яким користується комерсанти, є мобільним то він жорстко не прив'язана до конкретного місця роботи. Для коректної роботи системи візуалізації та формування списку товарів комерсанти зобов'язаний авторизуватися в системі і вибрати доступний склад зі списку. Після цього він може приступати до роботи на вибраному складі, але тільки якщо в систему доданий план розміщення товарів для цього складу. В іншому випадку система пошуку шляху працювати не буде,

Серверна частина системи відповідає за з'єднання з базою даних, перевірку даних, що надходять від клієнта, запис і виведення даних з бази даних та інші маніпуляції з даними. У момент звернення клієнтської частини до серверної відбувається валідація даних, вибірка або запис даних відповідно до запиту.

Якщо на клієнтській частині відбулася активація подія сканування накладної, то перед видачею інформації з серверної частини на клієнтську, дані передаються в модуль адаптивного пошуку оптимального маршруту. Всередині модуля дані сортуються, обробляються і відбувається пошук оптимального маршруту. Після формування масиву проходження шляху, дані передаються на клієнтську частину. Усередині клієнтської частини сформований список точок фіксується на модулі користувацького виведення. Це дозволяє формувати статичні карти маршрутів, щоб комерсанти міг приблизно орієнтуватися всередині складу.

Детальна робота модуля, що відповідає за побудову оптимального і адаптивного пошуку шляху надана на рисунку 4.

На блок-схемі представлений процес формування списку товарів, зазначених у накладній. Для цього комерсанти сканує через додаток спеціальний QR код на накладній. Інформація зі списком товарів завантажується через інтернет на пристрій. Після завантаження товару відбувається пошук самого далекого товару. Пошук маршруту формується шляхом прийняття за точку початку робоче місце комерсанти, а в якості мети вказується товар зі списку. Відбувається перебір всіх товарів в списку і формується новий відсортований масив з товарами з огляду на відстань до них. Комерсанти починає свій шлях з найдалшої точки і поступово просувається до свого робочого місця.

У разі, якщо комерсанти змінює свій маршрут прямування, система перебудовує маршрут згідно з внесеними змінами. Перевірка на проходження по маршруту відбувається в момент сканування товару, для занесення інформації в базу. Якщо товар що відсканували не відповідає товару, який був у списку система починає перебудовувати маршрут.

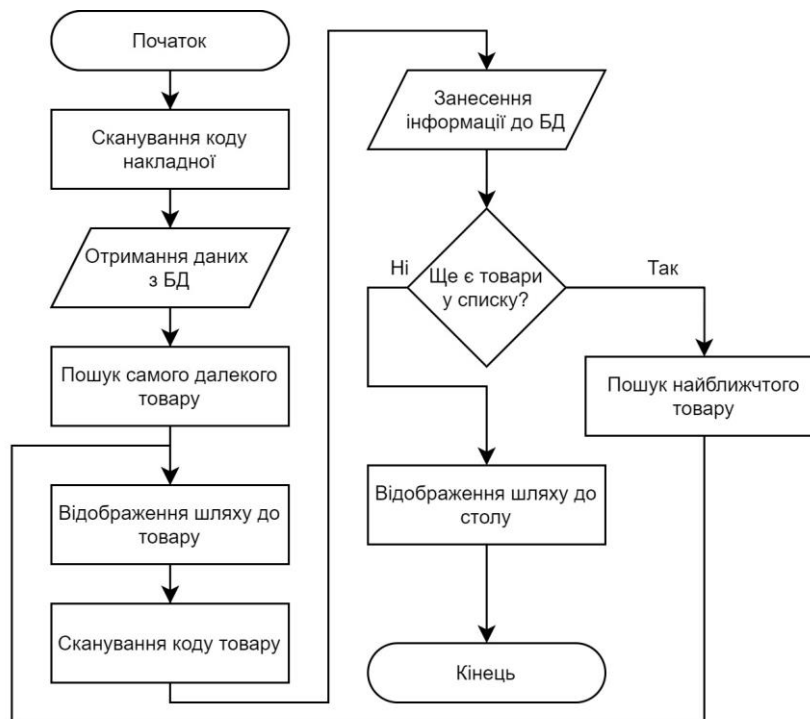


Рисунок 4 - Блок-схема стандартної роботи модуля пошуку шляху

### Результати експериментів

Для перевірки ефективності роботи системи адаптивного пошуку оптимального маршруту комірника для видачі товарів за накладною було проведено кілька експериментів. В якості випробуваного виступав комірник з досвідом роботи 1 місяць, на складі що тестувався. Для чистоти експерименту всі накладні буде збирати один випробуваний. Розмір складу становить 500м<sup>2</sup>. На складі розміщено 1300 різних видів продукції. Для експерименту був проведений збір товарів по 10 накладним.

У першому варіанті комірник виробляв збірку по 10 накладним із загальною кількістю позицій 50. У момент виконання накладних комірник покладався на свій особистий досвід. Розташування товарів вказувалося в форматі «номер стелажа» «номер комірки». Таким чином комірник сам вирішував куди йому потрібно зараз піти і який товар взяти. Час проходження записувався кожен раз, коли комірник віддавав товари користувачеві згідно з накладною.

У другому варіанті експерименту комірник також виробляв збір по 10 накладним із загальною кількістю позицій 50. Але в цей раз він слідував маршрутом, який йому формувала система оптимального пошуку маршруту. Час завершення виконання замовлення так само фіксувався в момент видачі товарів за накладною.

За результатами двох експериментів комірник виконав однакову кількість накладних з однаковою кількістю позицій. Так як накладні були ідентичними, то загальна відстань має залишитися незмінною. В обох випадках комірник рухався приблизно з однаковою швидкістю. Таким чином показники обох експериментів вийшли досить достовірними. Загальні показники двох експериментів представлені на рисунку 5.

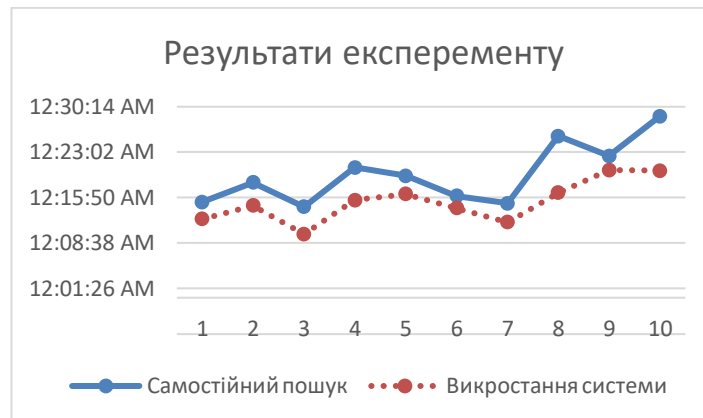


Рисунок 5 - Результати експерименту

За результатами експерименту видно, що при використанні системи адаптивного пошуку оптимального маршруту комірник витрачає менше часу на виконання накладної ніж при самостійному пошуку шляху. Середній час виконання однієї накладної при самостійній побудові маршруту склав 19 хвилин і 34 секунди. А при використанні системи загальний час знизився на 22% і склав 15 хвилин і 13 секунд. Отже, загальний час на виконання 10 накладних теж знизився на 22% і склав економію в 43 хвилини. Таким чином використання адаптивної системи оптимального пошуку маршруту для комірника є більш ефективним, ніж стандартні способи побудови шляху які використовує комірник.

#### Висновки та перспективи подальшого дослідження

Для формування адаптивного пошуку оптимального маршруту комірника для видачі товарів за накладною була розроблена система з формуванням карт розміщення осередків і алгоритмом зворотного пошуку маршруту. Завдяки розробленій системі, комірник витрачає менше часу на виконання замовлення за накладною, тому що йому не треба продумувати маршрут заповнення. Комірникові досить зробити сканування накладної за допомогою мобільного пристрою, а додаток самостійно побудує оптимальний маршрут.

Надалі планується провести модернізацію і доопрацювання системи шляхом обліку габаритів товару в момент формування маршруту. Таким чином комірник вироблятиме заповнення кошика, орієнтуючись на габарити продукції.

#### Список бібліографічного опису

1. WMS (warehouse management system, система управління складом) [Електронний ресурс]. - <https://www.it.ua/ru/knowledge-base/technology-innovation/warehouse-management-system-wms> (Дата звернення: 17.08.2020);
2. Lawler EL, Lenstra JK, Kan AR, Shmoys DB The Travelling Salesman Problem: A Guided Tour of Combinational Optimization. Vol. 3. New York, NY, Wiley, 1985. 476 p .;
3. Левітін А. В. Жадібні методи: алгоритм Дейкстри // Алгоритми. Введення в розробку і аналіз. - М.: Вільямс, 2006. - 576 с .;
4. Афанасьєва, Т. В. Алгоритми і програми: навчальний посібник / Т.В. Афанасьєва, Ю.С. Кувайскова, В.А. Фасхутдінова. - Ульяновськ: УлГТУ, 2011. - 227 с.
5. Вимоги до ТСД[Електронний ресурс]. -<https://rodden.ru/1680-trebovanija-dlja-tsd> (Дата звернення: 17.08.2020);
6. Smyth Neil. Android Studio 3.0 Development Essentials. Android 8 Edition, Payload Media, Inc., 2017. - 864 p.
7. Келер А., Бредскі Г. Вивчаємо OpenCV 3 - Learning OpenCV 3. -М.: ДМК-Пресс, 2017. - 826 с.

#### References

1. WMS (warehouse management system) [Electronic resource]. -<https://www.it.ua/ru/knowledge-base/technology-innovation/warehouse-management-system-wms> (Application date: 17.08.2020);
2. Lawler EL, Lenstra JK, Kan AR, Shmoys DB The Travelling Salesman Problem: A Guided Tour of Combinational Optimization. Vol. 3. New York, NY, Wiley, 1985. 476 p .;
3. Levitin AV Greedy methods: Dijkstra's algorithm // Algorithms. Introduction to development and analysis. - M .: Williams, 2006. - 576 p .;
4. Afanasieva, TV Algorithms and programs: a textbook / TV Afanasieva. Afanasyeva, Yu.Ye. Кувайскова, В.А. Fashutdinov. - Ulyanovsk: UISTU, 2011. - 227 p.
5. Requirements for TSD [Electronic resource]. -<https://rodden.ru/1680-trebovanija-dlja-tsd> (Access date: 17.08.2020);
6. Smyth Neil. Android Studio 3.0 Development Essentials. Android 8 Edition, Payload Media, Inc., 2017. - 864 p.
7. Keller A., Bredsky G. We study OpenCV 3 - Learning OpenCV 3. -M .: DMK-Press, 2017. - 826 p.