

*Музильов Д. О., к.т.н., доц.; Гречененко В. А.*

## **ОСОБЛИВОСТІ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ НА МОВІ JAVA ДЛЯ РІШЕННЯ ЛОГІСТИЧНИХ ЗАДАЧ**

*Розроблено програму для логістичних рішень на прикладі маятникового маршруту для імітації процесу перевезення тарно-штучних сільськогосподарських вантажів.*

**Вступ.** На сьогоднішній день існує достатньо програм, які можуть допомогти розрахувати будь-які логістичні задачі. При цьому, більшість з них є комерційними. Зазвичай перелік майже усіх даних в них приймається в якості константи. Такий підхід не може у повній мірі врахувати випадковий характер процесів, що спостерігаються при реальних перевезеннях на практиці. Це в першу чергу, притаманно техніко-експлуатаційним показникам, які є визначальними при плануванні логістичного процесу на оперативних та стратегічних періодах. Тому результати, які отримуються за допомогою існуючих програмних продуктів в повній мірі не відображають усі аспекти сучасної логістики.

**Основна частина.** Саме тому з'явилася необхідність створити програмний продукт, який більш адаптований до реальних процесів, що відбуваються у сучасних транспортних технологіях. Така некомерційна програма для перевезень допоможе провести розрахунки з урахуванням наступних випадкових величин. Таких як : експлуатаційна швидкість руху та час простою автомобіля під навантаженням та розвантаженням [1-2]. В якості прикладу наведено програму, що моделює процес перевезення на прикладі маятникового маршруту тарно-штучного сільськогосподарського вантажу. Мова програмування на котрій був розроблений алгоритм та графічний інтерфейс – Java. Мова програмування Java обрана тому, що це мультиплатформенний спосіб створення спеціалізованого «софту». А це значить, що вона підходить для роботи на будь-якій операційній системі ПК та сучасних гаджетів.

Для успішного створення програми необхідно розробити алгоритм, який би враховував та обробляв усі параметри, що необхідні. Далі представлена блок-схема алгоритму програми та опис основних змінних (рисунок 1). З алгоритму видно, що спочатку програма потребує вводу початкових даних:

- час виїзду з АТП ( $t_{\text{выезд}}$ );
- нульовий пробіг від АТП до точки А ( $L_{o1}$ ) та від точки В до АТП ( $L_{o2}$ );
- вантажний пробіг від точки А до точки В ( $L_{eT}$ );
- кількість їздок ( $n_{\text{езд}}$ );
- номінальну вантажопідйомність автомобіля ( $q_n$ ).

Для імітації процесу перевезення сільськогосподарського вантажу тип кузова автомобіля не уточнювався. Це компенсується окремим розрахунком часу простою під навантаженням при кожному циклі вантажної їздки автомобіля.

Далі здійснюється генерація наступних випадкових величин як швидкість руху ( $V_i$ ) та час навантаження ( $t_p$ ) і розвантаження ( $t_n$ ), в результаті чого програма дає на виході час виїзду з АТП ( $t_{\text{АТП}}^{\text{уб}}$ ). Для опису випадкового характеру, обраних техніко-експлуатаційних показників використовувався нормальний закон розподілу. Вибір саме нормального закону розподілу обумовлений результатами попередніх досліджень [3]. При визначенні характеру зміни випадкової величини часу простою під навантаженням і розвантаженням в період збору врожаю раніше встановлено, що її розподілення відбувається за експонентною [1, 3]

Також за спеціальними формулами розраховується час прибуття до пункту А ( $t_{\text{пр}i}^A$ ), час виїзду з пункту А ( $t_{\text{уб}i}^A$ ), а також час прибуття ( $t_{\text{пр}i}^B$ ) та виїзду з пункту В ( $t_{\text{уб}i}^B$ ), і наостанок розраховується час прибуття до АТП ( $t_{\text{АТП}}^{\text{пр}i}$ ).

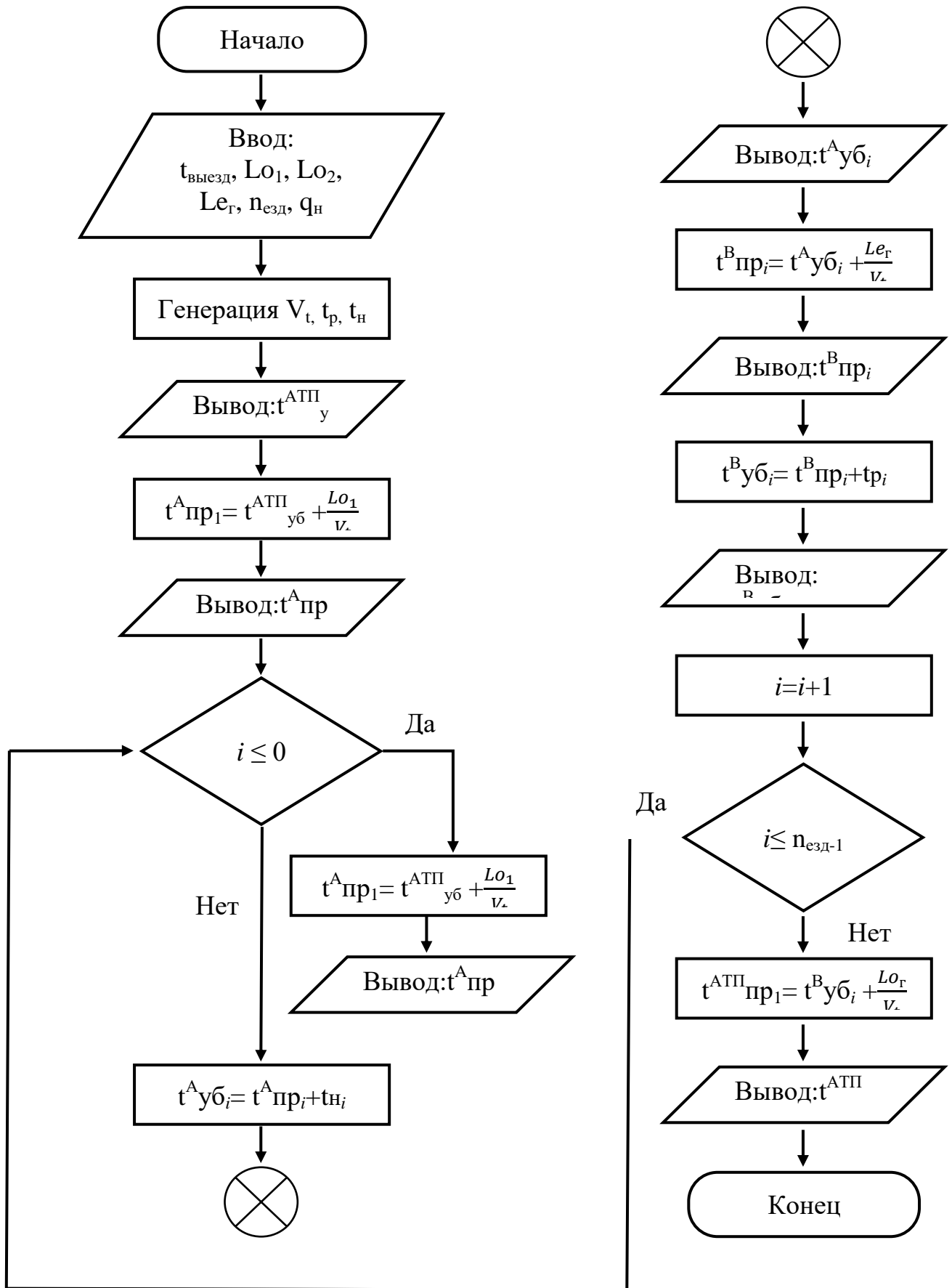


Рис.1 - Блок-схема алгоритм програми

Варто зазначити, що даний алгоритм робить розрахунок як для загального маршруту, так і для кожного окремого циклу з цього маршруту.

На рисунку 2 надано приклад лістингу програмного коду. Із рисунку видно, як виглядає реалізація алгоритму за допомогою сучасної мови Java.

```
String NEzd = nEzd.getText();
nEz = Double.parseDouble(NEzd);
String QN = qN.getText();
double tQN = Double.parseDouble(QN);
lPor = leG;
double UbATP = tDeparture;
double vT = 20 + (int)(Math.random() * ((30 - 20) + 1));
double APR = UbATP + ((lo1/vT)/60);
double tP = (10 + (int)(Math.random() * ((20 - 10) + 1)));
double UbA = APR + (tP/60);
double BPR = UbA + (leG / vT);
double tR = (15 + (int)(Math.random() * ((30 - 15) + 1)));
double UbB = BPR + (tR/60);
vT = 20 + (int)(Math.random() * ((30 - 20) + 1));
APR = this.UbB + (this.lPor / vT);
tP = 10 + (int)(Math.random() * ((20 - 10) + 1));
UbA = APR + (tP/60);
vT = 20 + (int)(Math.random() * ((30 - 20) + 1));
BPR = UbA + (this.leG / vT);
tR = (15 + (int)(Math.random() * ((30 - 15) + 1)));
for(i = 1; i<nEz; i++) {
    vT = 20 + (int)(Math.random() * ((30 - 20) + 1));
    APR = UbB + (lPor / vT);
    tP = 10 + (int)(Math.random() * ((20 - 10) + 1));
    UbA = APR + (tP/60);
    vT = 20 + (int)(Math.random() * ((30 - 20) + 1));
    BPR = UbA + (leG / vT);
    tR = (15 + (int)(Math.random() * ((30 - 15) + 1)));
    UbB = BPR + (tR/60);
}
vT = 20 + (int)(Math.random() * ((30 - 20) + 1));
double PRATP = UbB + (lo2/vT);
```

Рис. 2 - Зразок лістингу програми

Крім цього можна побачити зразки інтерфейсу розробленої нами програми на рис. 3 та 4.

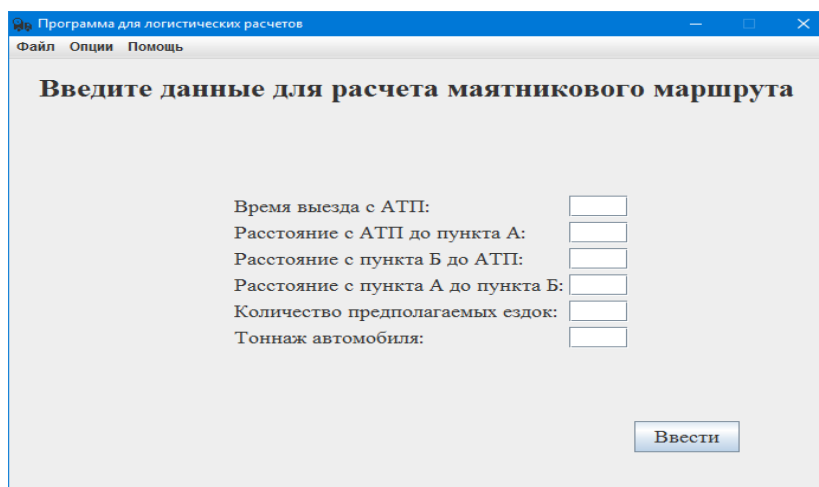


Рис. 3 - Початкове вікно програми з полями для надання вхідних параметрів

На рисунку 4 представлено результуючу таблицю з вихідними даними, яка відображає вихідні показники для кожного з циклів маршруту та загальні показники, тобто розклад руху.

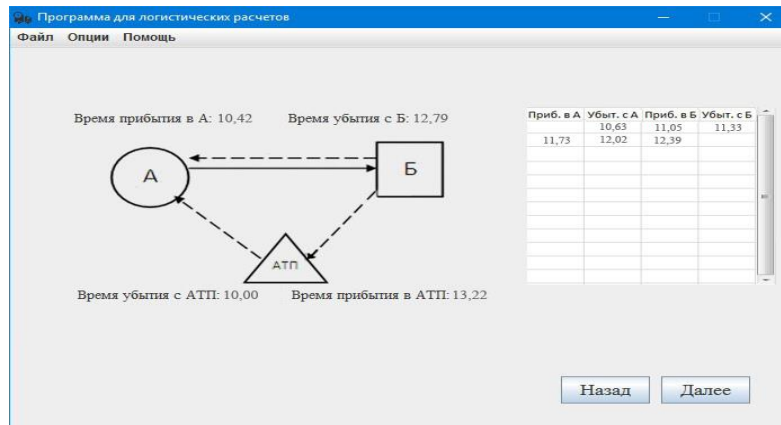


Рис.4 - Кінцеве вікно програми зі схемою маршруту та таблицею вихідних показників

За результатами роботи програми диспетчер отримує розклад роботи автомобіля на маршруті протягом робочої зміни. Інформація, що розраховано програмним продуктом легко інтегрується у стандартний пакет MS Office. Це збільшить рівень автоматизації процесу при обробці та аналізі звітних даних.

**Висновок.** Таким чином, розроблена програма має певні переваги серед існуючих на сьогодні програмних продуктів. Основною перевагою є те, що кінцевий варіант дозволяє створити докладний розклад необхідного маршруту з переліком випадкових величин, таких як: час навантаження та час розвантаження, експлуатаційну швидкість транспортного засобу, які в існуючих до цього програмах зазначені в якості констант, що обмежує в свою чергу можливість прийнята коректного рішення, при плануванні перевезень, користувачем.

#### Список літературних джерел

- Музылев Д. Критерий выбора рациональной технологии доставки сельскохозяйственных грузов [Текст] / Д. Музылев, Н. Карнаух, Н. Бережная, О. Кутья // Motrol. Commission of motorization and energetics in agriculture. – Vol. 17, №7. – Lublin – Rzeszow, 2015. – С. 67-73.
- Музылев Д. А. Определение рациональной структуры уборочно-транспортного комплекса [Текст] / Д. А. Музылев, Н. Г. Бережная // Научное обозрение. – 2015. – № 24. – С. 461-469.
- Музылев Д. А. Разработка методики выбора условий взаимодействия зерноуборочного и транспортного комплексов [Текст] / Д. А. Музылев, А. Г. Кравцов, Н. В. Карнаух, Н. Г. Бережная, О. В. Кутья // Східно-Європейський журнал передових технологій. – 2016. – Т. 2, № 3 (80). – С. 11-21. doi: 10.15587/1729-4061.2016.65670

**Музыльов Дмитро Олександрович** – к.т.н., доцент, доцент кафедри транспортних технологій і логістики, Харківський національний технічний університет сільського господарства імені Петра Василенка.

**Гречененко Владислав Артурович** – магістрант кафедри транспортних технологій і логістики, Харківський національний технічний університет сільського господарства імені Петра Василенка