

федеральное государственное бюджетное образовательное учреждение   
высшего образования   
«Санкт-Петербургский государственный технологический институт  
(технический университет)»  
СПбГТИ(ТУ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | Информатика и вычислительная техника | | | |
| Направление подготовки | | 09.03.03 | Прикладная информатика в химии | | | |
| Направленность (профиль) | |  | САПР | | | |
|  | |  |  | | | |
| Факультет | |  | Информационных технологий и управления | | | |
| Кафедра | |  | Систем автоматизированного  проектирования и управления | | | |
| Учебная дисциплина | |  | Методы поддержки принятия решений | | | |
|  |  | | |  |  |  |

Курс \_\_\_3\_\_\_ Группа 475

**Лабораторная работа №1**

**Транспортная задача**

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Андрианова К.И.\_\_

(подпись, дата) (инициалы, фамилия)

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Овчинников Р.С.

(подпись, дата) (инициалы, фамилия)

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пекер В.А.

(подпись, дата) (инициалы, фамилия)

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Уланов В.Н.

(подпись, дата) (инициалы, фамилия)

Санкт-Петербург

2019

# Цель работы

Целью данной работы является создание программы, которая осуществляет решение транспортной задачи с помощью алгоритма северо-западного угла. Также предусматривается разработка интерфейса для пользователя, с возможностью менять условия задачи.

**Ход выполнения работы**

Методы линейного программирования:

Линейное программирование представляет собой методы решения определенного класса задач по нахождению крайних значений (mах или min). В модели линейного программирования выделяются три составные части: целевая (максимизируемая или минимизируемая) функция, система ограничений и условие не отрицательности переменных.

Линейное программирование основано на решении системы линейных уравнений (с преобразованием в уравнения и неравенства), когда зависимость между изучаемыми явлениями строго функциональна. Для него характерны математическое выражение переменных величин, определенный порядок, последовательность расчетов (алгоритм), логический анализ. Применять его можно только в тех случаях, когда изучаемые переменные величины и факторы имеют математическую определенность и количественную ограниченность, когда в результате известной последовательности расчетов происходит взаимозаменяемость факторов, когда логика в расчетах, математическая логика совмещаются с логически обоснованным пониманием сущности изучаемого явления.

Для решения любой поставленной задачи в линейном программировании, нужно соблюсти 4-е правила:

* В задаче должен быть четко сформулирован и количественно определен **критерий оптимальности,** что не так легко сделать на практике.
* Важной составной частью задачи линейного программирования являются **ограничения,** связанные с наличными ресурсами, потребностями или другими факторами.
* Линейное программирование предполагает выбор вариантов, и оно применимо только тогда, когда конкретные условия экономической задачи обусловливают эту **свободу выбора.**
* Модель должна содержать **только линейные уравнения или неравенства,** т.е. все переменные задачи должны быть в первой степени.

По характеру решаемых задач методы линейного программирования можно разбить на две группы.

* **Универсальные методы.** С их помощью могут решаться любые задачи линейного программирования. Самым распространенным из них является симплексный метод, предложенный Дж. Данцигом, метод разрешающих множителей, разработанный академиком Л. В. Канторовичем в 1939 г., примерно за 10 лет до его появления за рубежом.
* **Специальные методы.** Эти методы проще универсальных, но применимы не для всех задач. К ним относятся распределительный метод для решения транспортной задачи, метод разрешающих слагаемых А. Л. Лурье, метод дифференциальных рент А. Л. Брудно, венгерский метод.
* К особой группе методов линейного программирования относятся **приближенные методы**, отличающие от остальных тем, что не гарантируют строго оптимального решения задачи, но они просты и хорошо приспособлены к ручным вычислениям. К ним относятся индексный метод, метод аппроксимации Фогеля, метод Северо-Западного угла и др.

Методы линейного программирования применяются для решения многих экстремальных задач, с которыми довольно часто приходится иметь дело в экономике. Решение таких задач сводится к нахождению крайних значений (максимума и минимума) некоторых функций переменных величин.

Одна из самых распространенных и востребованных оптимизационных задач в логистике – транспортная задача. В классическом виде она предполагает нахождение оптимального (т.е. сопряженного с минимальными затратами) плана грузоперевозок.

Решить транспортную задачу можно различными методами, начиная от симплекс-метода и простого перебора, и заканчивая методом графов.

**Транспортная задача (классическая)** — задача об [оптимальном](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B5) плане перевозок однородного продукта из однородных пунктов наличия в однородные пункты потребления на однородных транспортных средствах (предопределённом количестве) со статичными данными и линеарном подходе (это основные условия задачи).

Для классической транспортной задачи выделяют два типа задач: критерий стоимости (достижение минимума затрат на перевозку) или расстояний и критерий времени (затрачивается минимум времени на перевозку). Под названием транспортная задача, определяется широкий круг задач с единой математической моделью, эти задачи относятся к задачам линейного программирования и могут быть решены оптимальным методом. Однако, спец. метод решения транспортной задачи позволяет существенно упростить её решение, поскольку транспортная задача разрабатывалась для минимизации стоимости перевозок.

Два самых распространенных алгоритма для решения таких задач это алгоритм северо-западного угла и алгоритм минимального элемента.

1.Алгоритм северо - западного угла

Заполнение таблицы транспортной задачи начинается с левого верхнего угла, поэтому и называется метод северо-западного угла.

Метод состоит из ряда однотипных шагов, на каждом из которых, исходя из запасов очередного поставщика и запросов очередного потребителя, заполняется только одна клетка и соответственно исключается из рассмотрения один поставщик или один потребитель.

2. Алгоритм минимального элемента

Как и метод северо-западного угла, он состоит из ряда однотипных шагов, на каждом из которых заполняется только одна клетка таблицы, соответствующая минимальной стоимости, и исключается из рассмотрения только одна строка (поставщик) или один столбец (потребитель). Поставщик исключается из рассмотрения, если его запасы груза использованы полностью. Потребитель исключается из рассмотрения, если его запросы удовлетворены полностью. На каждом шаге исключается либо один поставщик, либо один потребитель.

В лабораторной работе для решения транспортной задачи мы используем метод Северо-Западного угла.

Суть метода: Метод состоит в последовательном переборе строк и столбцов транспортной таблицы, начиная с левого столбца и верхней строки, и выписывании максимально возможных отгрузок в соответствующие ячейки таблицы так, чтобы не были превышены заявленные в задаче возможности поставщика или потребности потребителя. На цены доставки в этом методе не обращают внимание, поскольку предполагается дальнейшая оптимизация отгрузок.

Алгоритм решения:  
**1. Определение исходного опорного решения (метод северо- западного угла).**

Пусть мы имеем таблицу исходных данных задачи. Исходное опорное решение будем строить, по так называемому, правилу «северо-западного угла».

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ai | bk | b1 | b2 | … | bk | … | bq |  |  |  |
| a1 | x11 | c11 | x12 | c12 | … | x1k | c1k | … | x1q | c1q |
|  |  |  |  |  |  |  |  |  |  |  |
| a2 | x21 | c21 | x22 | c22 | … | x2k | c2k | … | x2q | c2q |
|  |  |  |  |  |  |  |  |  |  |  |
| … | … | … | … | … | … | … |  |  |  |  |
| ai | xi1 | ci1 | xi2 | ci2 | … | xik | cik | … | xiq | ciq |
|  |  |  |  |  |  |  |  |  |  |  |
| … | … | … | … | … | … | … |  |  |  |  |
| ap | xp1 | cp1 | xp2 | cp2 | … | xpk | cpk | … | xpq | cpq |

Заполним вышеуказанную таблицу, начиная с левого верхнего угла, двигаясь далее или по строке вправо, или по столбцу вниз.

В клетку (1, 1) занесем меньшее из чисел***a1 и b1*,** т.е. ***x11****=min {****a1 , b1****}.*Если***a1 > b1 ,*** то ***x11****=****b1*** и первый столбец «закрыт», т. е. потребности первого потребителя удовлетворены полностью.

Двигаемся далее по первой строке, записывая в соседнюю клетку (1, 2) меньшее из чисел ***a1 - b1*** и ***b2*** т.е. ***x12****=min {****a1- b1 , b2****}.*

Если же ***b2*** > ***a1***, то аналогично «закрывается» первая строка и далее переходим к заполнению соседней клетки (2, 1), куда заносим ***x21****=min{****а2,b1 - a1****}.*

Будем продолжать этот процесс до тех пор, пока на каком-то этапе не исчерпываются ресурсы ***аp*** и потребности ***bq****.*

**Выбор инструментария**

Программа написана в Visual Studio 2017 на языке C#.

**Интерфейс**

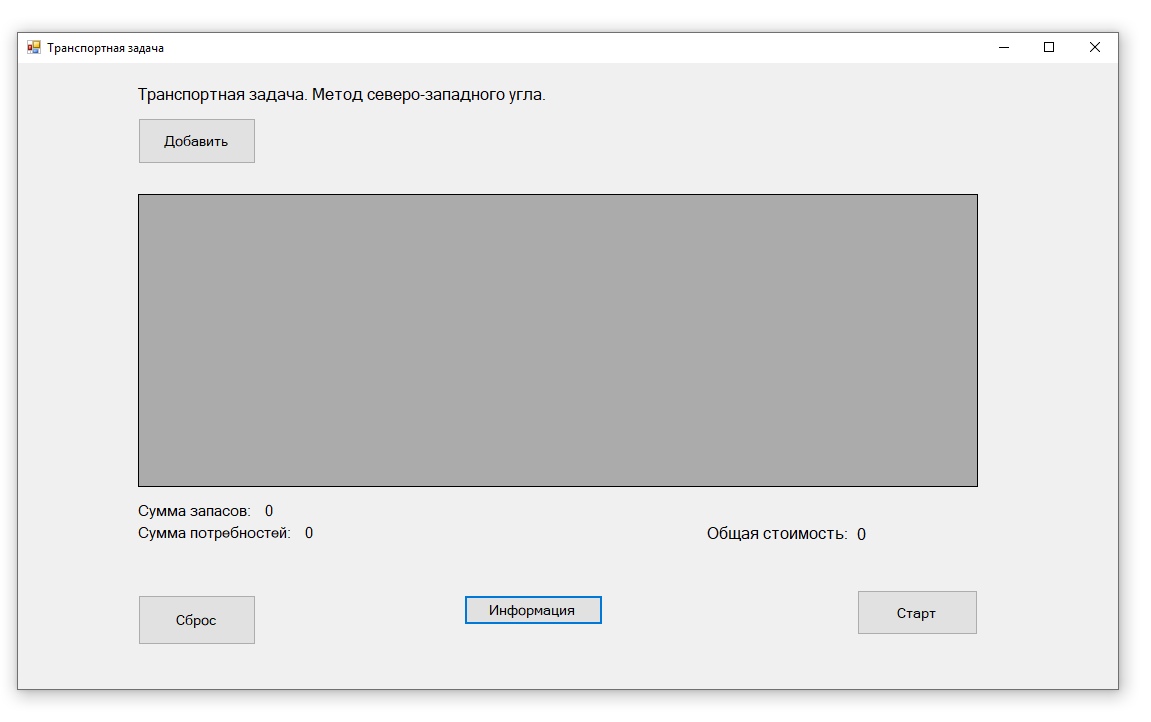
[](https://vk.com/photo59961986_457246663)

Рисунок 1- Интерфейс программы

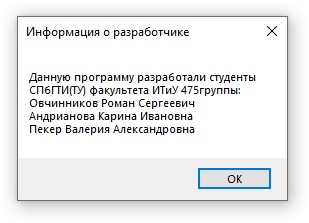
[](https://vk.com/photo59961986_457246664)

Рисунок 2- Информация о разработчиках

**Блок-схемы программы**

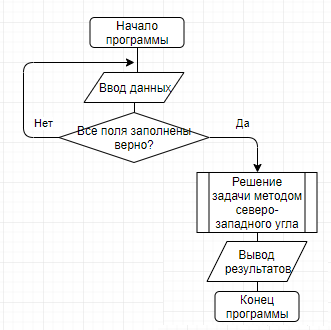


Рисунок 3 – Блок-схема программы

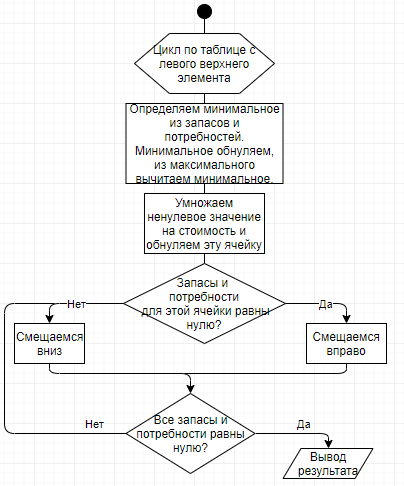


Рисунок 4- Блок-схема алгоритма северо-западного угла

**Тестирование**

Сначала мы заполняем таблицу с названием и количеством запасов каждого из потребителей (см рис.5). Далее заполняем таблицу с названием и количеством запасов поставщиков (см рис.6). После этого вводится коэффициенты стоимости запасов (см рис.7). Затем по указанным параметрам на экран выводится таблица стоимостей (см рис.8).

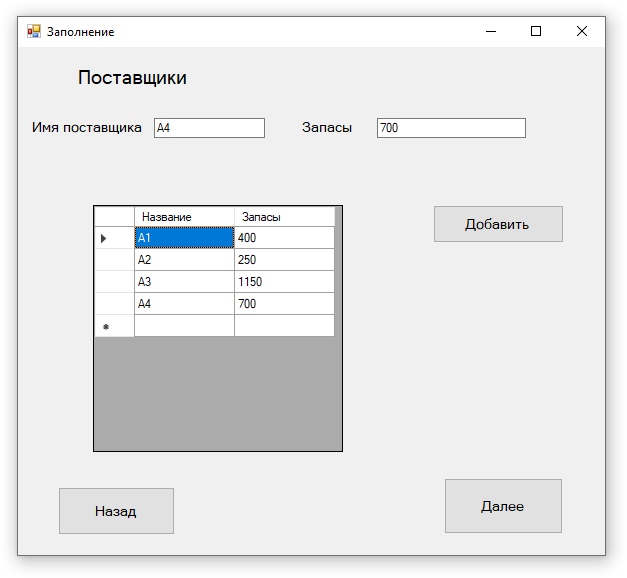
[](https://vk.com/photo59961986_457246662)

Рисунок 5-Ввод потребителей

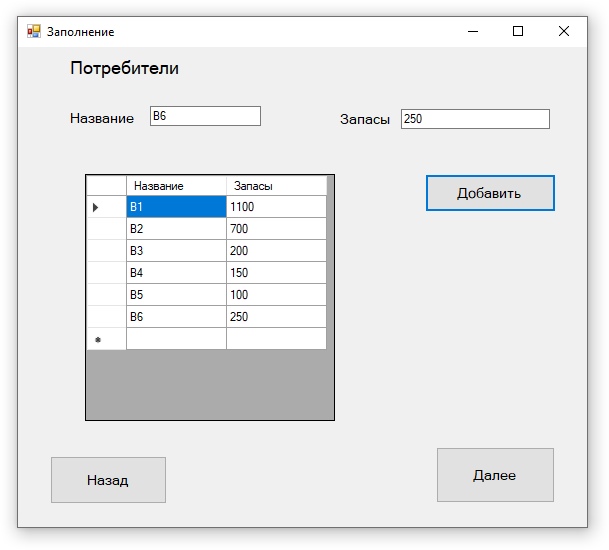
[](https://vk.com/photo59961986_457246661)

Рисунок 6-Ввод поставщиков

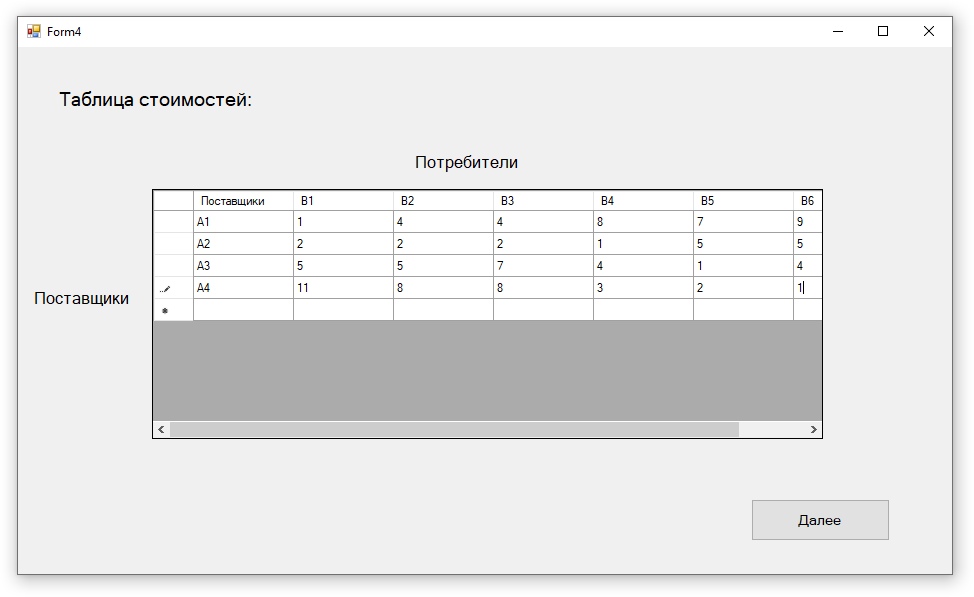
[](https://vk.com/photo59961986_457246660)

Рисунок 7- Ввод коэффициентов стоимости

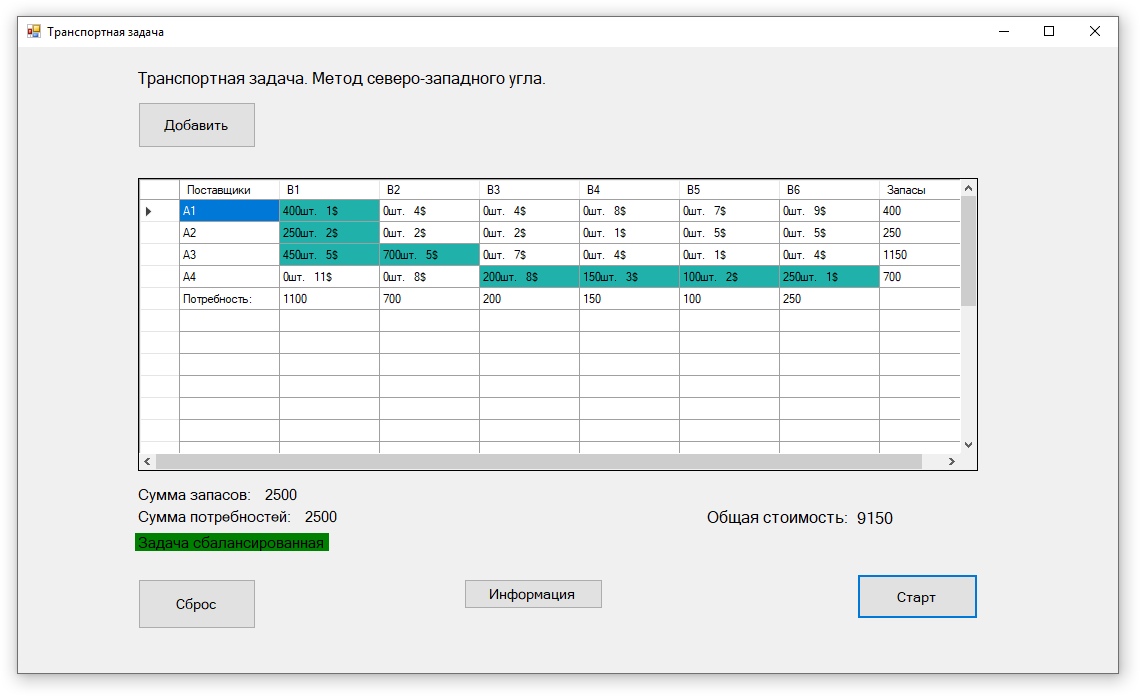
[](https://vk.com/photo59961986_457246659)

Рисунок 8-Таблица стоимостей

Далее мы проводим проверку алгоритма в случае одинакового числа запасов каждого склада и потребностей каждого потребителя (см рис.9,10).

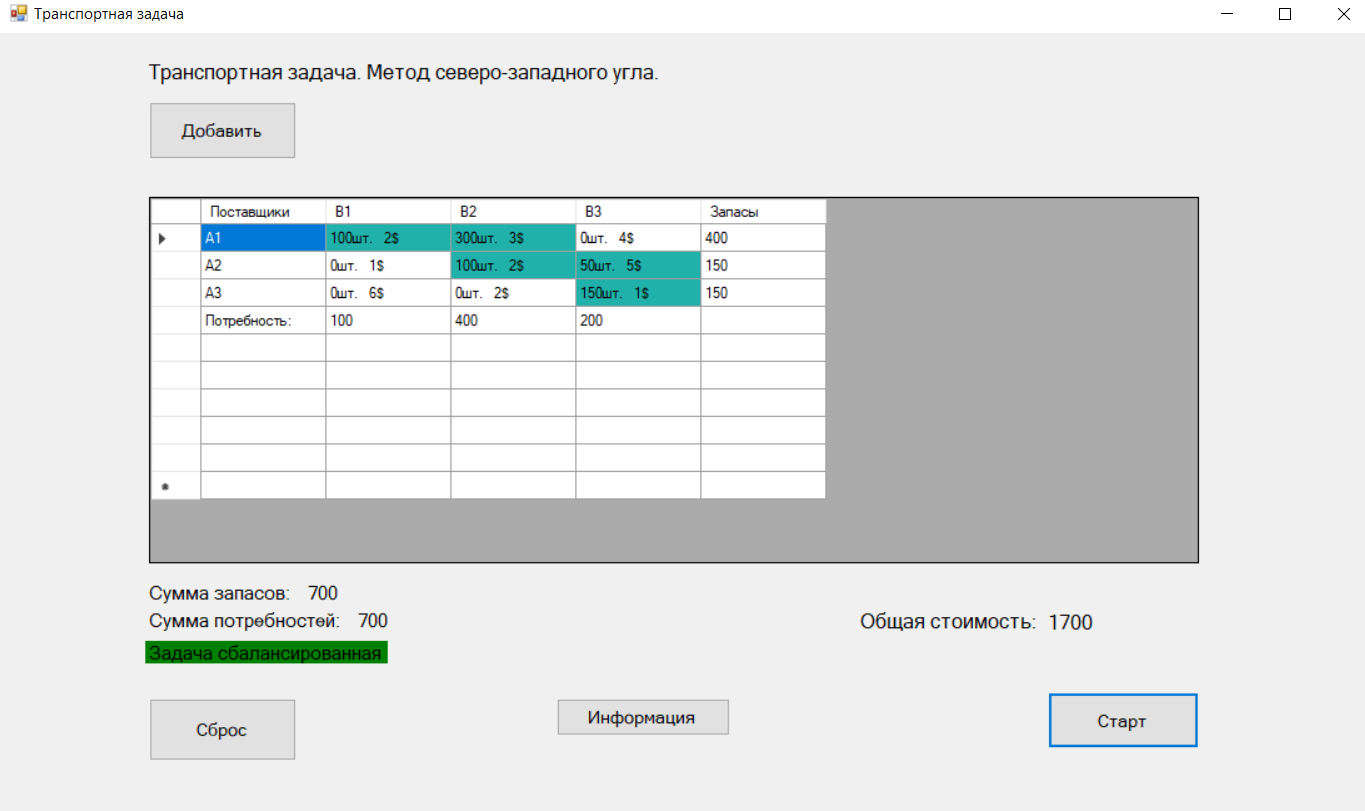


Рисунок 9- Проверка алгоритмов при   
одинаковом количестве запасов и потребностей

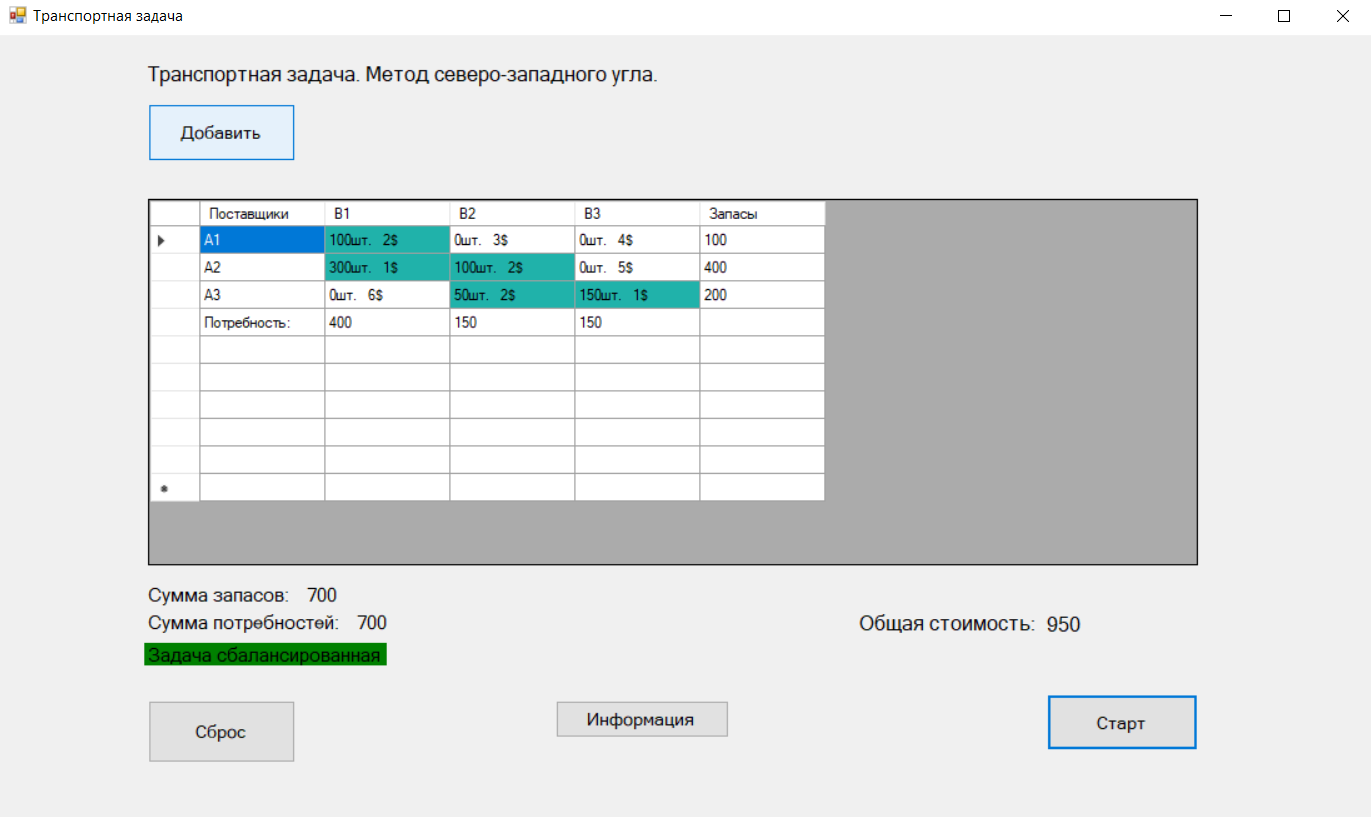


Рисунок 19- Проверка алгоритмов при   
одинаковом количестве запасов и потребностей

**Анализ тестирования**

В ходе тестирования мы убедились, что написанная нами программа работает. Весь функционал программы исправен и работает в полном режиме.

Метод северо-западного угла работает не эффективно, так как для метода важно расположение данных в матрице, также он не является самым оптимальным по цене и нуждается в усовершенствовании.

**Код программы (алгоритм)**

public string NordWest(int [,] matrix)

{

nordwest = "";

int x = 0;

int y = 0;

int sum = 0;

int[,] array = new int[3, 3];

for(int i = 0; i < 3; i++)

{

for (int j = 0; j <3; j++)

{

x = matrix[j,3];

y = matrix[3, i];

array[j,i] = Math.Min(x, y);

sum += array[j, i] \* matrix[j, i];

nordwest += array[j, i] + "\*" + matrix[j, i] + "+";

if (y - array[j,i] != 0)

{

matrix[3, i] -= array[j, i];

}

else if (y - array [j,i] == 0)

{

matrix[j, 3] -= array[j, i];

if (i +1 <3)

{

i++;

}

if (j-1 > -1)

{

j--;

}

}

}

}

nordwest = nordwest.Substring(0, nordwest.Length - 1);

nordwest += "=" + sum;

return nordwest;

}

**Вывод**

В ходе лабораторной работы мы изучили методы линейного программирования, ознакомились с транспортными задачами, методом северо-западного угла. Написали программу методом северо-западного угла для транспортной задачи. Используемый метод не является эффективным. Для северо-западного метода важно расположение данных в матрице, также он не является самым оптимальным по цене и нуждается в усовершенствовании.