

**ТВОРЧЕСКАЯ РАБОТА**

по дисциплине \_\_\_\_\_ Математическое моделирование \_\_\_\_\_  
\_\_\_\_\_ 5 семестр \_\_\_\_\_  
Тема работы \_\_\_\_\_ Решение Транспортных задач линейного  
Программирования. \_\_\_\_\_

Разработчик \_\_\_\_\_ Студент гр. \_\_\_\_\_ РП-31 \_\_\_\_\_

\_\_\_\_\_ Е.О. Карасёва \_\_\_\_\_

Руководитель \_\_\_\_\_ Преподаватель \_\_\_\_\_  
\_\_\_\_\_ Ю.С. Андрианова \_\_\_\_\_

**2019**

## Задание

на разработку индивидуального проекта  
по дисциплине «Математическое моделирование»

Тема задания: Транспортная задача.

Содержание пояснительной записки:

**Введение** (указать, краткую характеристику области применения и классификацию математических моделей. Актуальность использования математических моделей.)

### 1 Общий раздел

#### 1.1 Цель разработки

1.2 Характеристика математических методов (математическое обоснование моделей и методов решения данной задачи)

1.3 Описание инструментальных средств (характеристика инструментов программных средств исследования математических моделей)

1.4 Описание среды программирования (общая характеристика ИСР, описание используемых компонентов, их свойств, событий, методов)

### 2 Специальный раздел

#### 2.1 Постановка задачи (условие задачи)

2.2 Решение задачи (математическая модель, решение задачи с использованием математических методов и инструментальных средств)

2.3 Инструкция пользователя (описание последовательности действий пользователя, обеспечивающих загрузку, запуск, выполнение и завершение программы)

**Заключение** (вывод о проделанной работе)

**Приложение А** Решение задачи с использованием ИС.

**Приложение Б** Листинги текстов модулей проекта.

**Приложение В** Диск, содержащий ПЗ, проект со всеми модулями

## Введение.

Моделирование — это замещение некоторого объекта А другим объектом В. Замещаемый объект называется оригиналом, замещающий — моделью. Таким образом, модель — это заместитель оригинала. В зависимости от цели замещения модель одного и того же оригинала может быть различной. В науке и технике основной целью моделирования является изучение оригинала при помощи более простой его модели. Замещение одного объекта другим имеет смысл только в случае их определенного сходства, аналогии.

Математическая модель является приближенным, выраженным в математических терминах, представлением объектов, концепций, систем или процессов. Объекты, концепции, системы или процессы, подлежащие моделированию, называют объектами моделирования (ОМ).

Имитационные модели реализуются на ЭВМ в виде моделирующих алгоритмов (программ), позволяющих вычислять значения выходных переменных и определять новое состояние, в которое переходит модель при заданных значениях входных переменных, параметров и исходного состояния модели. Имитационное моделирование в отличие от численного характеризуется независимостью моделирующего алгоритма от типа информации, которую необходимо получить в результате моделирования. Достаточно универсальной, гибкой и эффективной является математическая модель, которая представляется в абстрактной математической форме посредством переменных, параметров, уравнений и неравенств.

Математические модели классифицируют по следующим критериям: 1) поведению моделей во времени; 2) видам входной информации, параметров и выражений, составляющих математическую модель; 3) структуре математической модели; 4) типу используемого математического аппарата.

Применительно к интегральным схемам можно предложить следующую классификацию.

В зависимости от характера свойств интегральной схемы математические модели делятся на функциональные и структурные.

Функциональные модели отображают процессы функционирования объекта, эти модели имеют форму систем уравнений.

По методу получения функциональные математические модели делятся на теоретические и формальные.

Теоретические модели получаются на основе изучения физических закономерностей, причем структура уравнений и параметры моделей имеют четкое физическое обоснование.

Формальные модели получаются при рассмотрении свойств реального объекта как черного ящика.

Сфера практического применения метода моделирования ограничивается возможностями и эффективностью формализации экономических проблем и ситуаций, а также состоянием информационного, математического, технического обеспечения используемых моделей. Стремление во что бы то ни стало применить математическую модель может не дать хороших результатов из-за отсутствия хотя бы некоторых необходимых условий.

Моделирование широко используется в науке и на производстве. Роль знаковых моделей особенно возросла с расширением масштабов применения ЭВМ при построении знаковых моделей. Современная форма «материальной реализации» знакового (прежде всего, математического) моделирования - это моделирование на цифровых электронных вычислительных машинах, универсальных и специализированных.

## Общий раздел.

### 1.1 Цель разработки.

Решить транспортную задачу при помощи метода Северо-Западного угла. Создать программу, которая облегчит решение данной задачи.

### 1.2 Характеристика математических методов (математическое обоснование моделей и методов решения данной задачи)

#### **Метод Северо-Западного угла.**

Методов формирования опорного плана в транспортной задаче придумано немало, но, пожалуй, самый простой из них – это метод Северо-Западного угла. Алгоритм заполнения клеток транспортной таблицы в его случае сводится к следующему: сначала заполняется клетка в верхнем левом угле («северо-западном» углу), затем следующая клетка справа и т.д., пока не заполнится вся строка. Затем мы переходим ко второй строке и снова заполняем ее слева направо. И так далее. Метод Северо-Западного угла, в самом деле, прост и понятен, но его недостаток – низкая эффективность. Сформированный с его помощью план в большинстве случаев не является оптимальным

#### **Метод Минимального Элемента.**

Один из методов составления опорного плана перевозок в транспортной задаче называется методом минимального элемента. Или, как его еще называют, метод наименьшей стоимости. Отличаясь простотой данный метод все же эффективнее чем, к примеру, метод Северо-западного угла. Кроме того, метод минимального элемента (или, иначе "метод наименьшего элемента") понятен и логичен. Его суть в том, что в транспортной таблице сначала заполняются ячейки с наименьшими тарифами, а потом уже ячейки с большими тарифами. То есть мы выбираем перевозки с минимальной стоимостью доставки груза. Это очевидный и логичный ход. Правда он не всегда приводит к оптимальному плану...

### 1.3 Описание инструментальных средств (характеристика инструментов программных средств исследования математических моделей)

#### **Решение транспортной задачи с помощью MS Excel.**

- 1) Ввести 2 диапазона с указанием поставщиков и потребителей: 1 диапазон – стоимость, 2 диапазон – количество груза, который изначально приравнивают к 0.
- 2) В отдельную ячейку вводим формулу для функции (каждая ячейка стоимости умножается на кол-во груза и всё суммируется)
- 3) Для каждого ограничения на поставщика или потребителя вводим формулу в отдельных ячейках.

Выполняем команду «Данные», «Поиск Решения», в появившемся окне устанавливаем нужные параметры.

Параметр: установить целевую ячейку (указать ссылку на ячейку с функцией).

Необходимо установить пункт к **минимальному значению**. В пункте изменяя ячейки, указываем ссылку на диапазон с грузом.

Добавляем на поставщиков и потребителей.

## Язык программирования C#

**C#, C-sharp, си-шарп** - язык программирования, сочетающий объектно-ориентированные и аспектно-ориентированные концепции. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств вроде Visual Studio.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML. Переняв многое от своих предшественников — языков C++, Java, Delphi, Модула и Smalltalk — C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++).

### Особенности языка

C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает FCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем. (Однако эта закономерность была нарушена с выходом C# 3.0, представляющим собой расширения языка, не опирающиеся на расширения платформы .NET.) CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.

### Название языка

Символ # в названии языка можно интерпретировать и как две пары плюсов ++, намекающие на новый шаг в развитии языка по сравнению с C++ (подобно шагу от C к C++), и как музыкальный символ диэз, вместе с буквой C составляющий в английском языке название ноты до-диэз. Последнее и дало название языку. Несмотря на то, что символ # (октоторп) в действительности является символом для обозначения номера на большинстве клавиатур и отличается от символа диэз # (Unicode U+266F), Microsoft, как автор языка, неоднократно обращалась к своим клиентам с просьбой принять это допущение.

### Стандартизация

Первая версия C# стандартизирована в ECMA (Standard ECMA-334 C# Language Specification, 3rd edition (June 2005)) и ISO (ISO/IEC 23270:2003, Information technology — C# Language Specification). Базируясь на этой спецификации, известно, как минимум, о двух независимых реализациях C#, находящихся в настоящее время на различных стадиях разработки

Mono, начата компанией Ximian и продолжена ее покупателем и преемником Novell.

В конце 2005 года Microsoft объявила о планах предать публичной спецификации вторую версию C#, но к настоящему времени эта спецификация еще не получила статус международного стандарта. Существующее положение ставит, таким образом, Microsoft в роль едва ли не единственной рыночной силы, развивающей C#.

## Версия 1.0

Проект C# был начат в декабре 1998 и получил кодовое название COOL (C-style Object Oriented Language). Версия 1.0 была анонсирована вместе с платформой .NET в июне 2000 года, тогда же появилась и первая общедоступная бета-версия; C# 1.0 окончательно вышел вместе с Microsoft Visual Studio .NET в феврале 2002 года.

Первая версия C# напоминала по своим возможностям Java 1.4, несколько их расширяя: так, в C# имелись свойства (выглядящие в коде как поля объекта, но на деле вызывающие при обращении к ним методы класса), индексаторы (подобные свойствам, но принимающие параметр как индекс массива), события, делегаты, циклы `foreach`, структуры, передаваемые по значению, автоматическое преобразование встроенных типов в объекты при необходимости (`boxing`), атрибуты, встроенные средства взаимодействия с неуправляемым кодом (DLL, COM) и прочее. Кроме того, в C# решено было перенести некоторые возможности C++, отсутствовавшие в Java: беззнаковые типы, переопределение операций (с некоторыми ограничениями, в отличие от C++), передача параметров в метод по ссылке, методы с переменным числом параметров, оператор `goto`. Также в C# оставили ограниченную возможность работы с указателями — в местах кода, специально обозначенных словом `unsafe` и при указании специальной опции компилятору.

## Версия 2.0

Проект спецификации C# 2.0 впервые был выложен Microsoft в октябре 2003 года; в 2004 году выходили бета-версии (проект с кодовым названием Whidbey), C# 2.0 окончательно вышел 7 ноября 2005 года вместе с Visual Studio 2005 и .NET 2.0.

### Новые возможности в версии 2.0

Частичные типы (разделение реализации класса более чем на один файл).

Обобщённые, или параметризованные типы (`generics`). В отличие от шаблонов C++, они поддерживают некоторые дополнительные возможности и работают на уровне виртуальной машины. Вместе с тем, параметрами обобщенного типа не могут быть выражения.

Новая форма итератора, позволяющая создавать сопрограммы с помощью ключевого слова `yield`, подобно Python'у и Руби.

Анонимные методы, обеспечивающие функциональность замыкания.

Оператор `'??'`: `return obj1 ?? obj2`; означает (в нотации C# 1.0) `return obj1 != null ? obj1 : obj2`.

Обнуляемые (`'nullable'`) типы-значения (обозначаемые вопросительным знаком, например, `int? i = null`); представляющие собой те же самые типы-значения, только могущие принимать также значение `null`. Такие типы позволяют улучшить взаимодействие с базами данных через язык SQL.

## Версия 3.0

В июне 2004 года Андерс Хейлсберг впервые рассказал на сайте Microsoft о планируемых расширениях языка в C# 3.0.[1] В сентябре 2005 года вышли проект спецификации C# 3.0 и бета-версия C# 3.0, устанавливаемая в виде дополнения к существующим Visual Studio 2005 и .NET 2.0.

### Новые возможности в версии 3.0

В C# 3.0 появились следующие радикальные добавления к языку:

ключевые слова `select`, `from`, `where`, позволяющие делать запросы из SQL, XML, коллекций и т. п. (запрос, интегрированный в язык, `Language Integrated Query`, или LINQ)

Инициализация объекта вместе с его свойствами:

`Customer c = new Customer(); c.Name = "James";`

превратится в

`Customer c = new Customer { Name = "James" };`

Лямбда-выражения:

`listOfFoo.Where(delegate(Foo x) { return x.size > 10; });`

превратится в

`listOfFoo.Where(x => x.size > 10);`

Подразумеваемый тип локальной переменной: `var x = "hello";` вместо `string x = "hello";`

Безымянные типы: `var x = new { Name = "James" };`

Методы-расширения — добавление метода в существующий класс с помощью ключевого слова `this` при первом параметре статической функции.

C# 3.0 будет совместим с C# 2.0 по генерируемому MSIL-коду; улучшения в языке — чисто синтаксические и реализуются на этапе компиляции. Например, многие из интегрированных запросов LINQ можно уже сейчас осуществить, используя безымянные делегаты в сочетании с предикатными методами над контейнерами вроде `List.FindAll` и `List.RemoveAll`.

## Специальный раздел

### 2.1 Постановка задачи (условие задачи)

2.4 Промышленный концерн имеет два завода и пять складов в различных регионах страны. Каждый месяц первый завод производит 40, а второй 70 ед. продукции. Вся продукция, производимая заводами, должна быть направлена на склады. Вместимость первого склада равна 20 ед. продукции; второго – 30; третьего – 15; четвертого – 27; пятого – 28 ед. Издержки транспортировки продукции от завода до склада следующие (ед.):

Заводы	Склады				
	1	2	3	4	5
1	520	480	650	500	720
2	450	525	630	560	750

Распределите план перевозок из условия минимизации ежемесячных расходов на транспортировку.

### 2.2 Решение задачи (математическая модель, решение задачи с использованием математических методов и инструментальных средств)

Классическая постановка транспортной задачи.

Пусть имеется  $m$  поставщиков  $A_1, A_2, \dots, A_i, \dots, A_m$  однородного груза в количествах соответственно  $a_1, a_2, \dots, a_i, \dots, a_m$  единиц и  $n$  потребителей  $B_1, B_2, \dots, B_i, \dots, B_n$  этого груза, потребность которых составляет соответственно  $b_1, b_2, \dots, b_i, \dots, b_n$  единиц.

Известны стоимости перевозок единицы груза от  $i$ -го поставщика к  $j$ -му потребителю —  $c_{ij}$  ( $i=1..m, j=1..n$ ).

Требуется составить такой план перевозок груза, который обеспечит минимальные транспортные расходы.

Замечание. Перевозимый груз должен быть однородным, например песок, уголь, лес, кирпичи, металл и т. п. Единицы измерения количества груза могут быть различными (т, шт., л и т. п.). Стоимость перевозки, как правило, измеряется в рублях. Предполагается, что стоимость перевозимого груза пропорциональна его количеству. В качестве поставщиков груза могут выступать предприятия, базы, склады, а в качестве потребителей — предприятия, магазины, строительные объекты и т. п.

Стоимость доставки единицы груза из каждого пункта отправления в соответствующие пункты назначения задана матрицей тарифов



	B1	B2	B3	B4	B5	Запасы
A1	520	480	650	500	720	40
A2	450	525	630	560	750	70
Потребности	20	30	15	27	28	

Проверим необходимое и достаточное условие разрешимости задачи.

$$\sum a = 40 + 70 = 110$$

$$\sum b = 20 + 30 + 15 + 27 + 28 = 120$$

Как видно, суммарная потребность груза в пунктах назначения превышает запасы груза на базах. Следовательно, модель исходной транспортной задачи является открытой. Чтобы получить закрытую модель, введем дополнительную (фиктивную) базу с запасом груза, равным 10 (110—120). Тарифы перевозки единицы груза из базы во все магазины полагаем равны нулю.

Занесем исходные данные в распределительную таблицу.

	B1	B2	B3	B4	B5	Запасы
A1	520	480	650	500	720	40
A2	450	525	630	560	750	70
A3	0	0	0	0	0	10
Потребности	20	30	15	27	28	

### Этап I. Поиск первого опорного плана.

1. Используя *метод наименьшей стоимости*, построим первый опорный план транспортной задачи.

Суть метода заключается в том, что из всей таблицы стоимостей выбирают наименьшую, и в клетку, которая ей соответствует, помещают меньшее из чисел  $a_i$  или  $b_j$ .

Затем, из рассмотрения исключают либо строку, соответствующую поставщику, запасы которого полностью израсходованы, либо столбец, соответствующий потребителю, потребности которого полностью удовлетворены, либо и строку и столбец, если израсходованы запасы поставщика и удовлетворены потребности потребителя.

Из оставшейся части таблицы стоимостей снова выбирают наименьшую стоимость, и процесс распределения запасов продолжают, пока все запасы не будут распределены, а потребности удовлетворены.

Искомый элемент равен  $c_{21}=450$ . Для этого элемента запасы равны 70, потребности 20. Поскольку минимальным является 20, то вычитаем его.

$$x_{21} = \min(70, 20) = 20.$$

x	480	650	500	720	40
450	525	630	560	750	70 - 20 = 50
x	0	0	0	0	10
20 - 20 = 0	30	15	27	28	

Искомый элемент равен  $c_{12}=480$ . Для этого элемента запасы равны 40, потребности 30. Поскольку минимальным является 30, то вычитаем его.

$$x_{12} = \min(40, 30) = 30.$$

x	<b>480</b>	650	500	720	<b>40 - 30 = 10</b>
450	x	630	560	750	50
x	x	0	0	0	10
0	<b>30 - 30 = 0</b>	15	27	28	

Искомый элемент равен  $c_{14}=500$ . Для этого элемента запасы равны 10, потребности 27. Поскольку минимальным является 10, то вычитаем его.  
 $x_{14} = \min(10, 27) = 10.$

x	480	x	<b>500</b>	x	<b>10 - 10 = 0</b>
450	x	630	560	750	50
x	x	0	0	0	10
0	0	15	<b>27 - 10 = 17</b>	28	

Искомый элемент равен  $c_{24}=560$ . Для этого элемента запасы равны 50, потребности 17. Поскольку минимальным является 17, то вычитаем его.  
 $x_{24} = \min(50, 17) = 17.$

x	480	x	500	x	0
450	x	630	<b>560</b>	750	<b>50 - 17 = 33</b>
x	x	0	x	0	10
0	0	15	<b>17 - 17 = 0</b>	28	

Искомый элемент равен  $c_{23}=630$ . Для этого элемента запасы равны 33, потребности 15. Поскольку минимальным является 15, то вычитаем его.  
 $x_{23} = \min(33, 15) = 15.$

x	480	x	500	x	0
450	x	<b>630</b>	560	750	<b>33 - 15 = 18</b>
x	x	x	x	0	10
0	0	<b>15 - 15 = 0</b>	0	28	

Искомый элемент равен  $c_{25}=750$ . Для этого элемента запасы равны 18, потребности 28. Поскольку минимальным является 18, то вычитаем его.  
 $x_{25} = \min(18, 28) = 18.$

x	480	x	500	x	0
450	x	630	560	<b>750</b>	<b>18 - 18 = 0</b>

x	x	x	x	0	10
0	0	0	0	28 - 18 = 10	

Искомый элемент равен  $x_{35}=0$ . Для этого элемента запасы равны 10, потребности 10. Поскольку минимальным является 10, то вычитаем его.

$$x_{35} = \min(10, 10) = 10.$$

x	480	x	500	x	0
450	x	630	560	750	0
x	x	x	x	0	10 - 10 = 0
0	0	0	0	10 - 10 = 0	

	B1	B2	B3	B4	B5	Запасы
A1	520	480[30]	650	500[10]	720	40
A2	450[20]	525	630[15]	560[17]	750[18]	70
A3	0	0	0	0	0[10]	10
Потребности	20	30	15	27	28	

В результате получен первый опорный план, который является допустимым, так как все грузы из баз вывезены, потребность магазинов удовлетворена, а план соответствует системе ограничений транспортной задачи

2. Подсчитаем число занятых клеток таблицы, их 7, а должно быть  $m + n - 1 = 7$ . Следовательно, опорный план является невырожденным.

Значение целевой функции для этого опорного плана равно:

$$F(x) = 480 \cdot 30 + 500 \cdot 10 + 450 \cdot 20 + 630 \cdot 15 + 560 \cdot 17 + 750 \cdot 18 + 0 \cdot 10 = 60870$$

### Этап II. Улучшение опорного плана.

Проверим оптимальность опорного плана. Найдем предварительные потенциалы  $u_i, v_j$  по занятым клеткам таблицы, в которых  $u_i + v_j = c_{ij}$ , полагая, что  $u_1 = 0$ .

$$u_1 + v_2 = 480; 0 + v_2 = 480; v_2 = 480$$

$$u_1 + v_4 = 500; 0 + v_4 = 500; v_4 = 500$$

$$u_2 + v_4 = 560; 500 + u_2 = 560; u_2 = 60$$

$$u_2 + v_1 = 450; 60 + v_1 = 450; v_1 = 390$$

$$u_2 + v_3 = 630; 60 + v_3 = 630; v_3 = 570$$

$$u_2 + v_5 = 750; 60 + v_5 = 750; v_5 = 690$$

$$u_3 + v_5 = 0; 690 + u_3 = 0; u_3 = -690$$

	$v_1=390$	$v_2=480$	$v_3=570$	$v_4=500$	$v_5=690$
$u_1=0$	520	480[30]	650	500[10]	720
$u_2=60$	450[20]	525	630[15]	560[17]	750[18]
$u_3=-690$	0	0	0	0	0[10]

Опорный план не является оптимальным, так как существуют оценки свободных клеток, для которых  $u_i + v_j > c_{ij}$

$$(2;2): 60 + 480 > 525; \Delta_{22} = 60 + 480 - 525 = 15 > 0$$

Выбираем максимальную оценку свободной клетки (2;2): 525

Для этого в перспективную клетку (2;2) поставим знак «+», а в остальных вершинах многоугольника чередующиеся знаки «-», «+», «-».

	1	2	3	4	5	Запасы
1	520	480[30][-]	650	500[10][+] 1	720	40
2	450[20]	525[+]	630[15]	560[17][-]	750[18]	70
3	0	0	0	0	0[10]	10
Потребности	20	30	15	27	28	

Цикл приведен в таблице  $(2,2 \rightarrow 2,4 \rightarrow 1,4 \rightarrow 1,2)$ .

Из грузов  $x_{ij}$  стоящих в минусовых клетках, выбираем наименьшее, т.е.  $y = \min(2, 4) = 17$ . Прибавляем 17 к объемам грузов, стоящих в плюсовых клетках и вычитаем 17 из  $x_{ij}$ , стоящих в минусовых клетках. В результате получим новый опорный план.

	B1	B2	B3	B4	B5	Запасы
A1	520	480[13]	650	500[27]	720	40
A2	450[20]	525[17]	630[15]	560	750[18]	70
A3	0	0	0	0	0[10]	10
Потребности	20	30	15	27	28	

Проверим оптимальность опорного плана. Найдем *предварительные потенциалы*  $u_i, v_j$  по занятым клеткам таблицы, в которых  $u_i + v_j = c_{ij}$ , полагая, что  $u_1 = 0$ .

$$u_1 + v_2 = 480; 0 + v_2 = 480; v_2 = 480$$

$$u_2 + v_2 = 525; 480 + u_2 = 525; u_2 = 45$$

$$u_2 + v_1 = 450; 45 + v_1 = 450; v_1 = 405$$

$$u_2 + v_3 = 630; 45 + v_3 = 630; v_3 = 585$$

$$u_2 + v_5 = 750; 45 + v_5 = 750; v_5 = 705$$

$$u_3 + v_5 = 0; 705 + u_3 = 0; u_3 = -705$$

$$u_1 + v_4 = 500; 0 + v_4 = 500; v_4 = 500$$

	$v_1=405$	$v_2=480$	$v_3=585$	$v_4=500$	$v_5=705$
$u_1=0$	520	480[13]	650	500[27]	720
$u_2=45$	450[20]	525[17]	630[15]	560	750[18]
$u_3=-705$	0	0	0	0	0[10]

Опорный план является оптимальным, так все оценки свободных клеток удовлетворяют условию  $u_i + v_j \leq c_{ij}$ .

Минимальные затраты составят:  $F(x) = 480 \cdot 13 + 500 \cdot 27 + 450 \cdot 20 + 525 \cdot 17 + 630 \cdot 15 + 750 \cdot 18 + 0 \cdot 10 = 60615$

**Анализ оптимального плана.**

Из 1-го склада необходимо груз направить в 2-й магазин (13 ед.), в 4-й магазин (27 ед.)

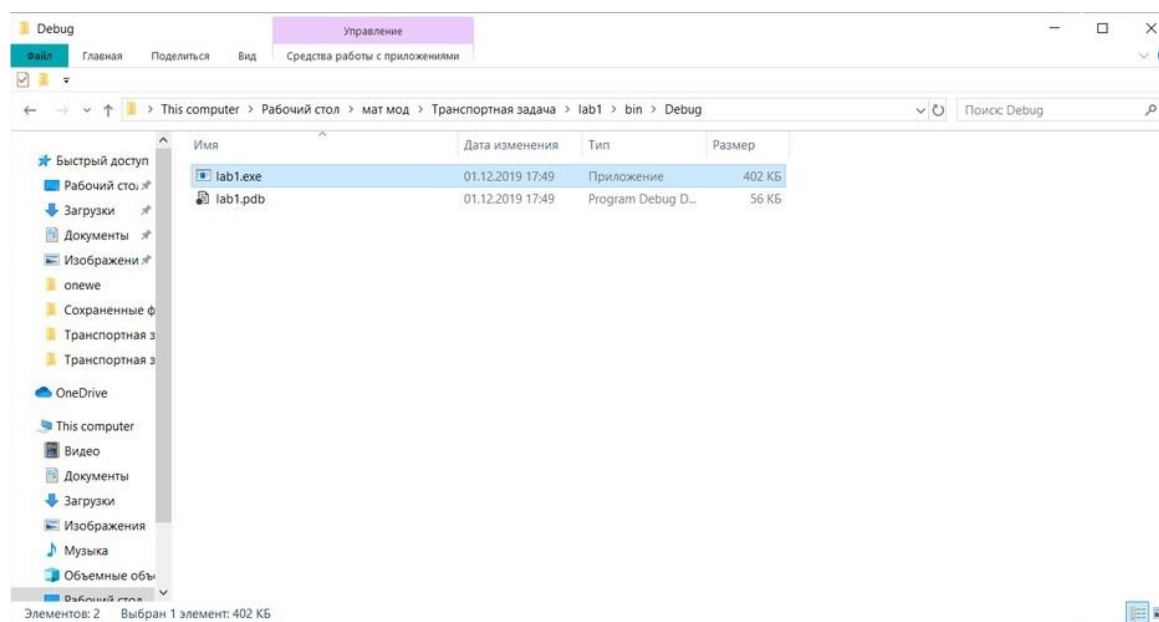
Из 2-го склада необходимо груз направить в 1-й магазин (20 ед.), в 2-й магазин (17 ед.), в 3-й магазин (15 ед.), в 5-й магазин (18 ед.)

Потребность 5-го магазина остается неудовлетворенной на 10 ед.

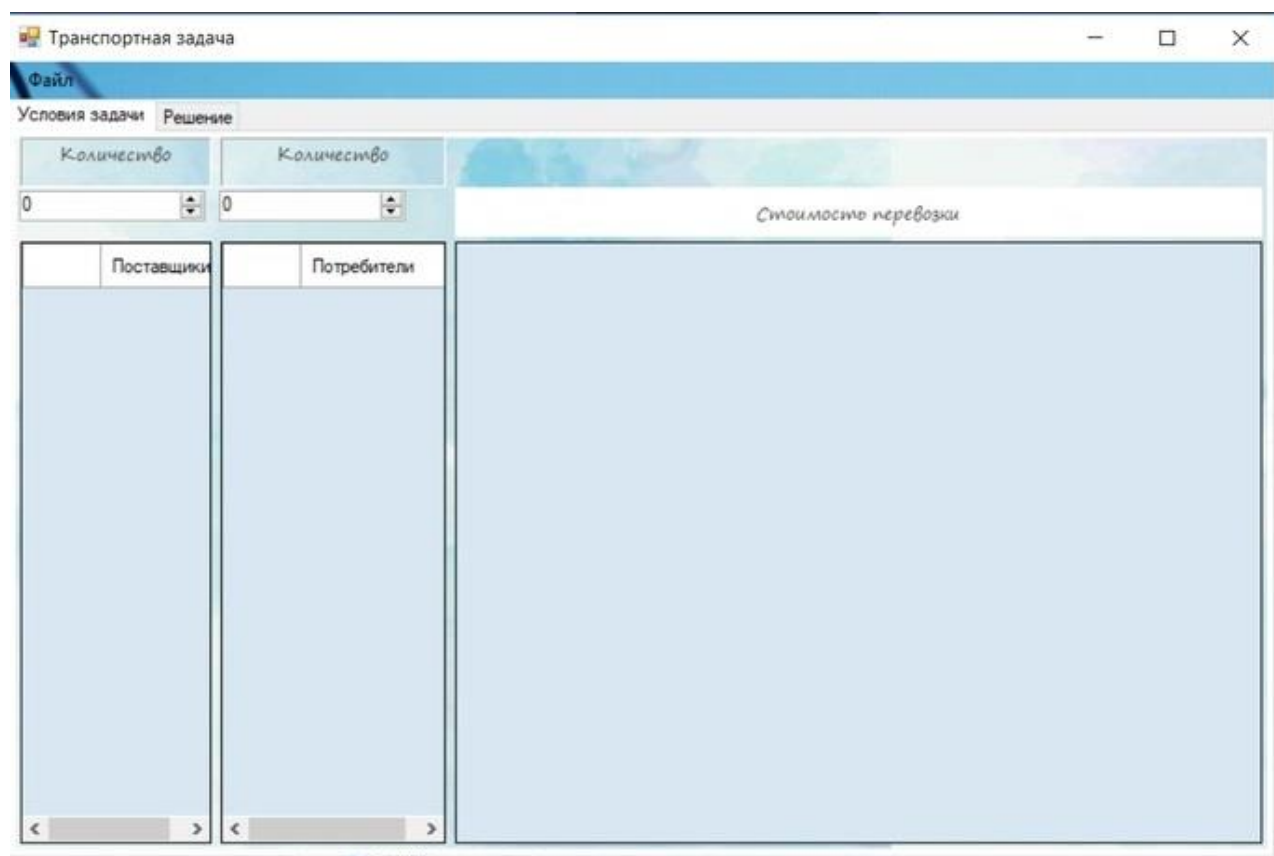
Оптимальный план является вырожденным, так как базисная переменная  $x_{35}=0$ .

## 2.3 Инструкция пользователя (описание последовательности действий пользователя, обеспечивающих загрузку, запуск, выполнение и завершение программы)

1. Открыть папку с программой.
2. Открыть папку lab1.
3. Открыть папку bin.
4. Открыть папку Debug.
5. Запустить lab1.exe.



6. Поставить количество поставщиков и потребителей.



Транспортная задача

Файл

Условия задачи    Решение

Количество    Количество

2    0

Стоимость перевозки

Поставщики		Потребители	
A1	0	A1	null
A2	0	A2	null

Транспортная задача

Файл

Условия задачи    Решение

Количество    Количество

2    5

Стоимость перевозки

Поставщики		Потребители				
A1	0	B1	0			
A2	0	B2	0			
		B3	0			
		B4	0			
		B5	0			

7. Ввести нужные значения в таблицы.

Транспортная задача

Файл

Условия задачи Решение

Количество 2 5

Стоимость перевозки

Поставщики		Потребители			B1	B2	B3	B4	B5
A1	40	B1	20	A1	520	480	650	500	720
A2	70	B2	30	A2	450	525	630	560	750
		B3	15						
		B4	27						
		B5	28						

8. Открыть вкладку «Решение».

Транспортная задача

Файл

Условия задачи Решение

Опорный план

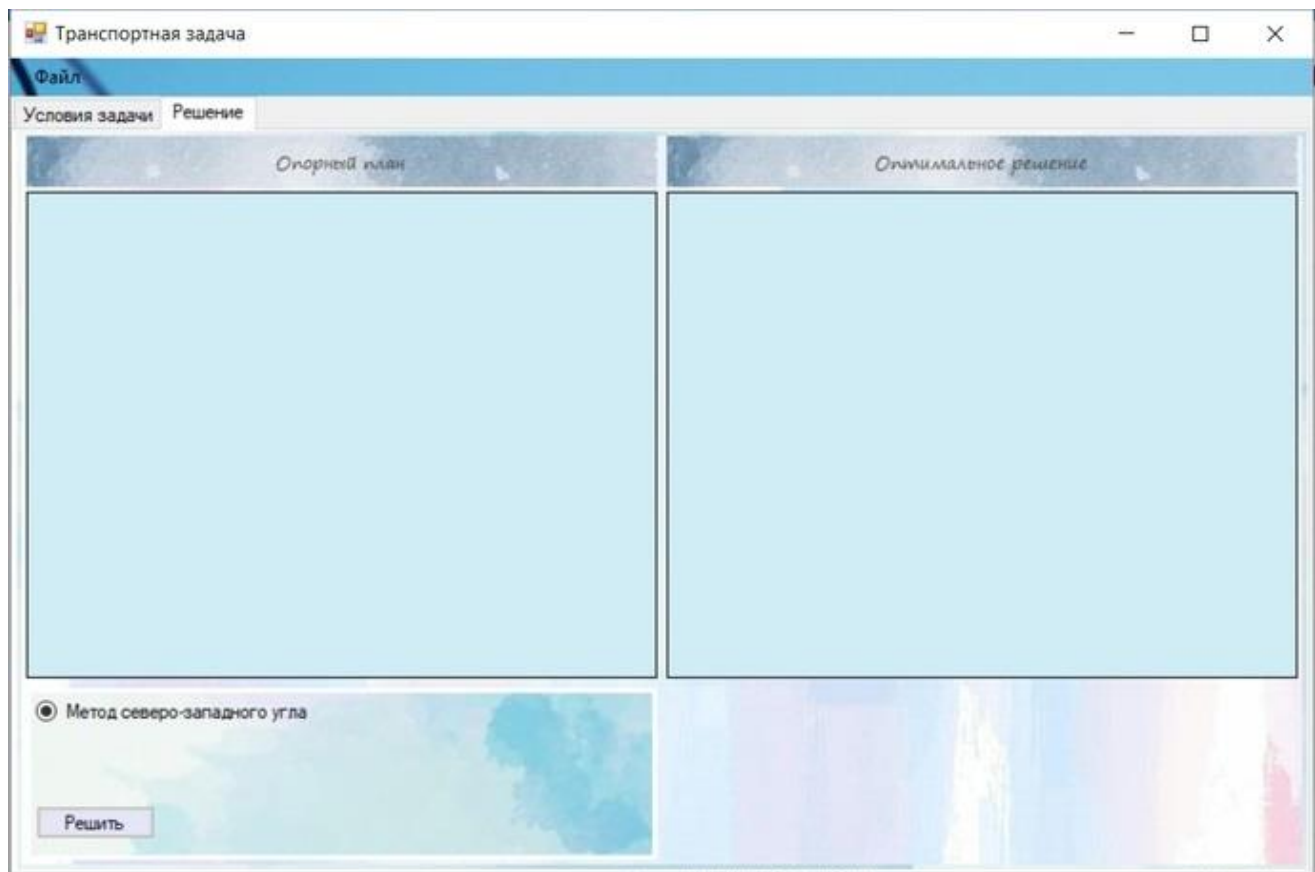
Оптимальное решение

Метод северо-западного угла

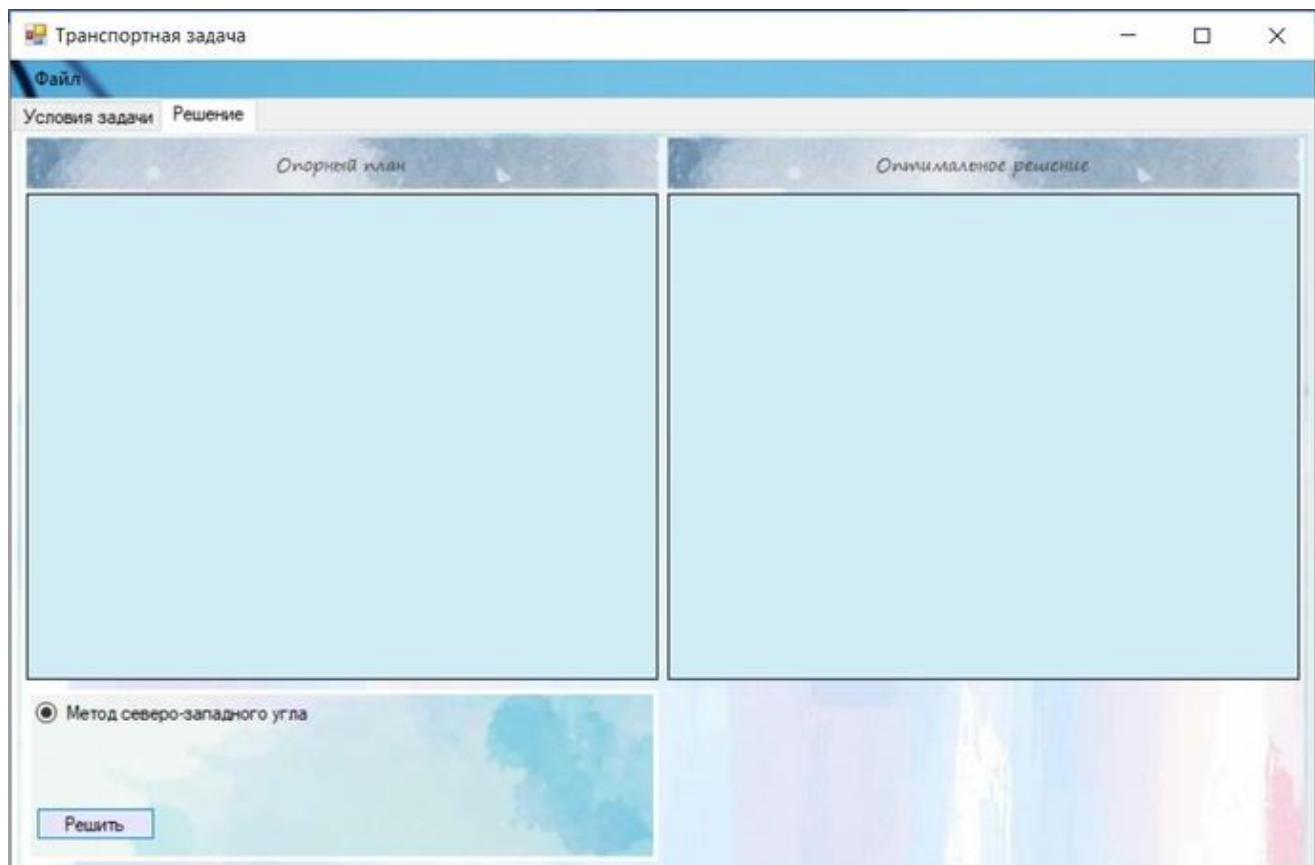
Решить

9. выбрать способ решение задачи. (метод северо-западного угла)

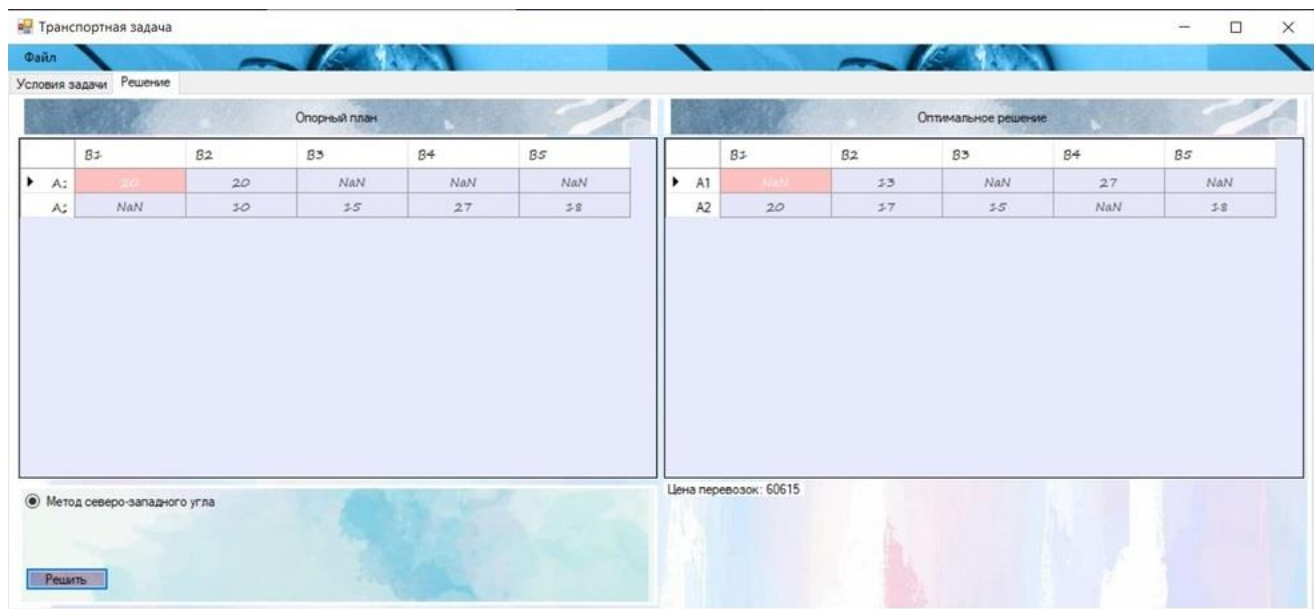




10. Нажать кнопку Решить.



11. Поздравляю, ваша задача решена!



### Заключение (вывод о проделанной работе)

При помощи языка C# была создана автоматизированная программа для решения Транспортных задач, тем самым облегчая решение любой Транспортной задачи для других пользователей. Так же было проведено подробное исследование решения транспортных задач, что может послужить опорным материалом для решения. А так же бы представлен подробный метод решения одной из них с учётом решения данной задачи в Excel.

### Приложение А Решение задачи с использованием ИС.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Поставки	1	2	3	4	5																	
2	A	520	480	650	500	720																	
3	B	450	525	630	560	750																	
4	C	0	0	0	0	0																	
5																							
6		0	13	0	27	0	40	40															
7		20	17	15	0	18	70	70															
8		0	0	0	0	10	10	10															
9		20	30	15	27	28																	
10		20	30	15	27	28																	
11																							
12	Функция		60615																				

B9									
	A	B	C	D	E	F	G	H	
1	Поставщик	1	2	3	4	5			
2	A	520	480	650	500	720			
3	B	450	525	630	560	750			
4	C	0	0	0	0	0			
5									
6		0	13	0	27	0	40	40	
7		20	17	15	0	18	70	70	
8		0	0	0	0	10	10	10	
9		20	30	15	27	28			
10		20	30	15	27	28			
11									
12	Функция		60615						

C12									
	A	B	C	D	E	F	G	H	I
1	Поставщик	1	2	3	4	5			
2	A	520	480	650	500	720			
3	B	450	525	630	560	750			
4	C	0	0	0	0	0			
5									
6		0	13	0	27	0	40	40	
7		20	17	15	0	18	70	70	
8		0	0	0	0	10	10	10	
9		20	30	15	27	28			
10		20	30	15	27	28			
11									
12	Функция		60615						
13									

G6									
	A	B	C	D	E	F	G	H	
1	Поставщик	1	2	3	4	5			
2	A	520	480	650	500	720			
3	B	450	525	630	560	750			
4	C	0	0	0	0	0			
5									
6		0	13	0	27	0	40	40	
7		20	17	15	0	18	70	70	
8		0	0	0	0	10	10	10	
9		20	30	15	27	28			
10		20	30	15	27	28			
11									
12	Функция		60615						
13									