МЕЖГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «????????????»

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РЕШЕНИЯ ТРАНСПОРТНОЙ ЗАДАЧИ ВЕНГЕРСКИМ МЕТОДОМ

Курсовая работа

по дисциплине «Системный анализ и исследование операций»

КР.1-53 01 02.10028412.15

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Самусев Д.А. АСОИ-181

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Плиско И. Г.

(подпись)

Дата допуска к защите \_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_

Могилев 2020

МЕЖГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Факультет Электротехнический

«Утверждаю»

Заведующий кафедрой\_\_\_\_\_\_\_\_\_\_

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.

З А Д А Н И Е

на курсовую работу по дисциплине «Системный анализ и исследование операций» студенту Самусев Д.А. гр. № АСОИ-181

1. Тема курсовой работы: Разработка приложения для решения транспортной задачи венгерским методом

2. Срок сдачи законченной работы: 18.05.2020

3. Исходные данные к курсовой работе:

Среда разработки: на выбор студента.

Разработать приложение для решения следующего типа задачи. Составить приложение для решения транспортной задачи венгерским методом.

4. Перечень подлежащих разработке вопросов:

Введение

1 Постановка задачи

2 Проектирование программного модуля

3 Реализация программного модуля

4 Тестирование программного модуля

Заключение

Список использованных источников

5. На проверку предоставляются пояснительная записка, исходные тексты

программ и исполняемые файлы на электронном носителе.

Руководитель курсовой работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Плиско И. Г.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Самусев Д. А.

Содержание

[Введение 3](#_Toc34772304)

[1 Постановка задачи 4](#_Toc34772305)

[1.1 Математическая модель задачи 4](#_Toc34772306)

[1.2 Входные данные 5](#_Toc34772307)

[1.3 Выходные данные 5](#_Toc34772309)

[1.4 Обработка ошибок 5](#_Toc34772311)

[2 Проектирование программного модуля 6](#_Toc34772313)

[2.1 Разработка структурной диаграммы программного модуля и её описание 6](#_Toc34772314)

[2.2 Разработка пользовательского интерфейса 6](#_Toc34772315)

[3 Реализация программного модуля 7](#_Toc34772317)

[3.1 Код программы 7](#_Toc34772318)

[4 Тестирование программного модуля 8](#_Toc34772319)

[Заключение 9](#_Toc34772320)

[Литература 10](#_Toc34772321)

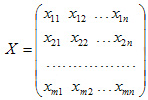
Введение

Под названием “транспортная задача” объединяется широкий круг задач с единой математической моделью. Классическая транспортная задача – задача о наиболее экономном плане перевозок однородного продукта или взаимозаменяемых продуктов из пунктов производства в пункты потребления, встречается чаще всего в практических приложениях линейного программирования. Линейное программирование является одним из разделов математического программирования – области математики, разрабатывающей теорию и численные методы решения многомерных экстремальных задач с ограничениями. Огромное количество возможных вариантов перевозок затрудняет получение достаточно экономного плана эмпирическим или экспертным путем. Применение математических методов и вычислительных в планировании перевозок дает большой экономический эффект. Транспортные задачи могут быть решены симплексным методом, однако матрица системы ограничений транспортной задачи настолько своеобразна, что для ее решения разработаны специальные методы. Эти методы, как и симплексный метод, позволяют найти начальное опорное решение, а затем, улучшая его получить оптимальное решение. В зависимости от способа представления условий транспортной задачи она может быть представлена в сетевой (схематичной) или матричной (табличной) форме. Транспортная задача может также решаться с ограничениями и без ограничений. В данной курсовой работе будут рассматриваться математическая постановка транспортной задачи линейного программирования - венгерский метод.

1 Постановка задачи

* 1. Математическая модель задачи

Переменными (неизвестными) транспортной задачи являются xij , i=1,2,...,m j=1,2,...,n - объемы перевозок от i-го поставщика каждому j-му потребителю. Эти переменные могут быть записаны в виде матрицы перевозок:



Так как произведение Cij\*Xij определяет затраты на перевозку груза от i-го поставщика j-му потребителю, то суммарные затраты на перевозку всех грузов равны:



По условию задачи требуется обеспечить минимум суммарных затрат.  
Следовательно, целевая функция задачи имеет вид:



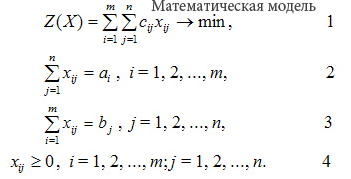
Система ограничений задачи состоит из двух групп уравнений. Первая группа из m уравнений описывает тот факт, что запасы всех m поставщиков вывозятся полностью и имеет вид:



Вторая группа из n уравнений выражает требование удовлетворить запросы всех n потребителей полностью и имеет вид:



Учитывая условие неотрицательности объемов перевозок, математическая модель выглядит следующим образом:



В рассмотренной модели транспортной задачи предполагается, что суммарные запасы поставщиков равны суммарным запросам потребителей, т.е.:



Такая задача называется задачей с **правильным балансом**, а модель задачи **закрытой**. Если же это равенство не выполняется, то задача называется задачей с **неправильным балансом**, а модель задачи — **открытой**. **Математическая формулировка транспортной задачи** такова: найти переменные задачи X=(xij), i=1,2,...,m; j=1,2,...,n, удовлетворяющие системе ограничений (цифра 2 на математической модели) , (3), условиям неотрицательности (4) и обеспечивающие минимум целевой функции (1).

* 1. Входные данные

Входными данными являются количество грузов для каждого транспорта, потребность каждой точки назначения и стоимость перевозки каждым транспортом в каждую точку назначения.

* 1. Выходные данные

Выходными данными являются значение целевой функции (минимальная стоимость перевозок) и матрица распределения транспортов по точкам назначения с необходимым количеством ресурсов.

* 1. Обработка ошибок

В процессе работы, приложение может работать некорректно при введении некорректных данных, таких как отрицательные или строковые значения в качестве входных данных. Для предотвращения подобных исключительных ситуаций разработан механизм, не позволяющий введение подобных данных.

2 Проектирование программного модуля

* 1. Разработка структурной диаграммы программного модуля и её описание
  2. Разработка пользовательского интерфейса

Пользовательский интерфейс представляет собой React js приложение, взаимодействующее с .Net приложением. Пользовательский интерфейс вначале принимает на вход размерность матрицы цен (количество строк и столбцов) - рисунок 1.

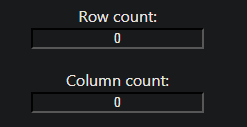


Рисунок 1

Затем заполняется матрица цен, массив имеющихся ресурсов (правый столбец) и массив потребностей (массив под матрицей) – рисунок 2.

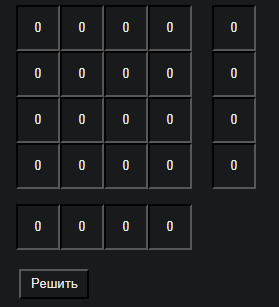


Рисунок 2

После нажатия кнопки “решить” будет предоставлено решение указанной задачи – рисунок 3.



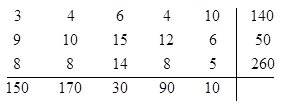
Рисунок 3

3 Реализация программного модуля

3.1 Код программы

4 Тестирование программного модуля

Для тестирования возьмём следующее условие:



Матрица цен имеет размерность 3x5. Введём данные соответствующие поля (рисунок 4).

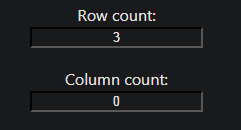


Рисунок 4

Далее заполним соответствующее данные в появившейся матрице (рисунок 5) и нажмём на кнопку “решить”.

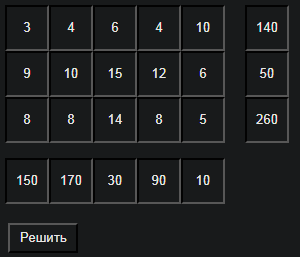


Рисунок 5

После нажатия на кнопку появляется матрица распределения ресурсов и значение целевой функции (рисунок 6).

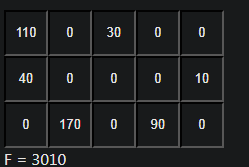
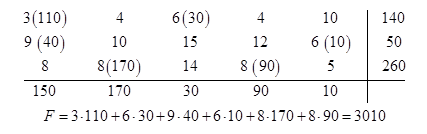


Рисунок 6

Сравним решение с предоставленным решением в примере:



Решение, предоставленное выше, совпадает с решением, которое предоставляет приложение. Таким образом, можно сделать вывод, что приложение работает корректно.

Заключение

В результате выполнения данной работы были закрепление основы и углублены знания приемов программирования на языке С#, получены практические навыки на всех этапах создания программного продукта.

В процессе выполнения работы решены следующие задачи: изучение и освоение паттерна проектирования MVC, внедрение HTML разметки в проект, предусмотрены возможные исключительные ситуации, разработана система аутентификации и авторизации в приложении.

В итоге был разработан веб-сервис позволяющий контролировать успеваемость студентов конкретного учебного заведения.

Литература

1. http://htmlbook.ru

2. https://docs.microsoft.com

3. http://professorweb.ru

4. https://metanit.com/sharp

5. https://stackoverflow.com

Приложение А

Startup.cs

**namespace** AccountingSystem