Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Магнитогорский государственный технический университет им. Г. И. Носова»

Многопрофильный колледж

**Отчет**

**по учебной практике**

**по специальности 09.02.07 Информационные системы и программирование**

**(базовой подготовки)**

**ПМ. 02. Осуществление интеграции программных модулей**

|  |  |
| --- | --- |
|  | Студента гр. ИСпПК-22-7  Гусев Денис Евгеньевич |
|  | Руководитель практики от МпК:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.В. Кобыльская  (дата, подпись) |

Магнитогорск, 2024

**ВНУТРЕННЯЯ ОПИСЬ**

**документов, находящихся в отчете**

Обучающегося гр. ИСпПК-22-7 Д.Е. Гусев

|  |  |  |
| --- | --- | --- |
| **№п/п** | **Наименование документа** | **Стр.** |
|  | Задание на практику | 3 |
|  | Аттестационный лист | 6 |
|  | Отчет о выполнении заданий по практике | 7 |

**ЗАДАНИЕ**

**на учебную практику**

**Обучающегося** гр. ИСпПК-22-7 Гусева Дениса Евгеньевича

**специальности** 09.02.07 Информационные системы и программирование

**по профессиональному модулю** ПМ. 02. Осуществление интеграции программных модулей, Учебная практика

**Цели практики:**

1. Получение практического опыта:

1. Разработка и оформление требований к программным модулям по предложенной документации.
2. Разработка тестовых наборов (пакетов) для программного модуля.
3. Разработка тестовых сценариев программного средства.
4. Инспектирование разработанных программных модулей на предмет соответствия стандартам кодирования.
5. Интеграция модулей в программное обеспечение.
6. Отладка программных модулей.

2. Формирование профессиональных компетенций (ПК)

|  |  |
| --- | --- |
| Код и наименование формируемых компетенций | Виды и объем производственных работ, выполняемых в период практики в рамках формируемых компетенций |
| ПК2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент  ПК2.2 Выполнять интеграцию модулей в программное обеспечение  ПК2.3 Выполнять отладку программного модуля с использованием специализированных программных средств  ПК2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения  ПК2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования | 1. Выработка требований к программному обеспечению и программному модулю. 2. Построение структуры программного продукта. 3. Проектирование программного продукта. 4. Внешнее проектирование (разработка внешней спецификации, разработка тестов) 5. Внутреннее проектирование (разработка схем проекта) 6. Разработка документа «Техническое задание» (разработка и оформление документа, согласование документа с заказчиком и руководителем, корректировка документа) 7. Структурное проектирование программных модулей 8. Проектирование программного обеспечения с использованием программных средств 9. Приведение программных средств в соответствие со стандартами 10. Разработка требовании к программному проекту 11. Определение необходимых ресурсов для создания программного продукта 12. Написание программного кода программного обеспечения. 13. Интеграция проектируемых модулей в компьютерную систему 14. Оценка надежности интегрируемых модулей 15. Отладка программных продуктов 16. Разработка тестовых наборов для заданного программного продукта 17. Проведения верификации программного продукта 18. Разработка пользовательской документации 19. Разработка руководства по применению 20. Постановка задачи. 21. Выбор математической модели. 22. Выбор и обоснование численного метода. 23. Разработка программы. 24. Решение модельных примеров. 25. Сопоставление с теорией. 26. Решение прикладной задачи. |

3. Формирование общих компетенций (ОК)

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.

ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.

ОК 9. Использовать информационные технологии в профессиональной деятельности.

ОК10.Пользоваться профессиональной документацией на государственном и иностранном языках.

ОК11.Планировать предпринимательскую деятельность в профессиональной сфере.

Место практики: ФГБОУ ВО «МГТУ им. Г.И. Носова», Многопрофильный колледж, полигон учебных практик*.*

**Задание на практику**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Содержание работ на практике** | **Примерные сроки выполнения (час)** |
|  | Разработка документа «Техническое задание» (разработка и оформление документа, согласование документа с заказчиком и руководителем, корректировка документа) | 6 |
|  | Постановка задачи. | 6 |
|  | Выбор математической модели. | 6 |
|  | Проектирование программного обеспечения с использованием программных средств | 12 |
|  | Внутреннее проектирование (разработка схем проекта) | 18 |
|  | Разработка программы. | 12 |
|  | Написание программного кода программного обеспечения | 12 |
|  | Разработка и модификация модулей программного обеспечения в соответствии с требованиями заказчика | 18 |
|  | Отладка программных модулей и выявление ошибок программного обеспечения | 12 |
|  | Разработка руководства по применению | 6 |

Примерный перечень документов, прилагаемых в качестве приложения к отчету по практике

1. Аттестационный лист по практике;
2. Характеристика
3. Дневник по практике

Руководитель практики от МпК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Кобыльская (*подпись)*

20 мая 2024 г.

**АТТЕСТАЦИОННЫЙ ЛИСТ ПО учебной ПРАКТИКЕ**

Гусева Дениса Евгеньевича

Обучающийся на 3 курсе специальности 09.02.07 Информационные системы и программирование успешно прошел практику по профилю специальности по профессиональному модулю: ПМ. 02. Осуществление интеграции программных модулей в организации: ФГБОУ ВО «МГТУ им. Г.И. Носова», Многопрофильный колледж, полигон учебных практик*.*

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| Код формируемых компетенций | Виды и объем работ, выполненных обучающимися во время практики в рамках формируемых компетенций | Оценка  зачтено/  не зачтено |
| ПК2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент  ПК2.2 Выполнять интеграцию модулей в программное обеспечение  ПК2.3 Выполнять отладку программного модуля с использованием специализированных программных средств  ПК2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения  ПК2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования | Разработка документа «Техническое задание» (разработка и оформление документа, согласование документа с заказчиком и руководителем, корректировка документа) |  |
| Постановка задачи. |  |
| Выбор математической модели. |  |
| Проектирование программного обеспечения с использованием программных средств |  |
| Внутреннее проектирование (разработка схем проекта) |  |
| Разработка программы. |  |
| Написание программного кода программного обеспечения |  |
| Разработка и модификация модулей программного обеспечения в соответствии с требованиями заказчика |  |
| Отладка программных модулей и выявление ошибок программного обеспечения |  |
| Разработка руководства по применению |  |

Руководитель практики от МпК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Кобыльская

(подпись)

1 июня 2024 г.

М.П.

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ ПО УЧЕБНОЙ ПРАКТИКЕ**

1. **Постановка задачи**
   1. **Описание условия задачи**

Задача заключается в составлении опорного плана и оптимизации перевозок с использованием метода потенциалов. Для этого необходимо создать и проанализировать клетки матрицы плана перевозок, разделяя их на базисные (заполненные) и свободные (незаполненные). Опорный план строится с помощью метода "северо-западного угла".

Для оптимизации плана перевозок необходимо вычислить потенциалы строк и столбцов, добавляется одна строка сверху и один столбец слева в матрицу плана перевозок. Потенциалы строки и столбца вычисляются по формуле ui + vj = cij, где ui и vj - это потенциалы строки и столбца соответственно, а cij - значение из базисной клетки.

* 1. **Выбор ввода данных (ручной/автоматический)**

С целью получение более точных расчетов был выбран ручной ввод данных, так как требуется анализировать и заполнять клетки матрицы перевозок, находить опорный план и вычислять потенциалы строк и столбцов. Такой способ позволит более детально и внимательно провести необходимые расчеты.

* 1. **Описание входных данных**

Входные данные включают в себя количество покупателей, продавцов и стоимость доставки. Далее формируется опорный план методом "северо-западного угла" для последующих автоматических расчетов.

* 1. **Описание выходных данных**

Выходные данные метода потенциалов включают оптимальный план перевозок, который содержит положительные числа cj и vj, называемые потенциалами строк и столбцов соответственно. После определения потенциалов строк и столбцов необходимо произвести перераспределение ресурсов между поставщиками и потребителям. В итоге получится оптимальный план перевозок с минимальной стоимостью перевозок.

1. **Построение алгоритма решения**
   1. **Построение словесного алгоритма**

1. Запросить данные у пользователя про количество покупателей, поставщиков и стоимость доставки.

2. Создать матрицу плана перевозок и добавить одну строку и один столбец к матрице плана перевозок.

3. Определить базисные клетки и свободные клетки.

4. Вычислить потенциалы строк и столбцов с использованием формулы ui + vj = сij.

5. Произвести оценку свободных клеток по формуле ui + .vj - cij ≤0

6. Найти наибольшую оценку в матрице свободных клеток.

* 1. **Построение блок-схемы**

1. **Программирование**
   1. **Написание кода программы**

Для решения поставленной задачи был написан код, представленный в листинге 1. В начале работы программы запрашивается количество покупателей, продавцов и цены, после чего автоматически строится опорный план, рассчитываются свободные клетки и находится клетка с наибольшей оценкой.

Листинг 1 – Код для построения опорного плана с оптимизацией

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication11

{

class Program

{

struct Element

{

public int Delivery { get; set; }

public int Value { get; set; }

public static int FindMinElement(int a, int b)

{

if (a > b) return b;

if (a == b) { return a; }

else return a;

}

}

static void Main(string[] args)

{

int i = 0;

int j = 0;

int ResultCoast = 0;

int max=0;

Console.Write("Введите количество поставщиков: ");

int n = Convert.ToInt32(Console.ReadLine());

int[] a = new int[n];

int[] copieda = new int[a.Length];

Console.Write("Введите количество покупателей: ");

int m = Convert.ToInt32(Console.ReadLine());

int[] b = new int[m];

int[] copiedb = new int[b.Length];

Element[,] C = new Element[n, m];

for (i = 0; i < a.Length; i++)

{

Console.Write($"Введите количество товара поставщика {i+1}: ");

a[i] = Convert.ToInt32(Console.ReadLine());

}

Array.Copy(a, copieda, a.Length);

for (j = 0; j < b.Length; j++)

{

Console.Write($"Введите спрос покупателя {j+1}: ");

b[j] = Convert.ToInt32(Console.ReadLine());

}

Array.Copy(b, copiedb, b.Length);

for (i = 0; i < n; i++)

{

for (j = 0; j < m; j++)

{

Console.Write($"Введите цену перевозки от продавца {i+1} до покупателя {j+1}: ");

C[i, j].Value = Convert.ToInt32(Console.ReadLine());

}

}

i = j = 0;

while (i < n && j < m)

{

try

{

if (a[i] == 0) { i++; }

if (b[j] == 0) { j++; }

if (a[i] == 0 && b[j] == 0) { i++; j++; }

C[i, j].Delivery = Element.FindMinElement(a[i], b[j]);

a[i] -= C[i, j].Delivery;

b[j] -= C[i, j].Delivery;

}

catch { }

}

Console.WriteLine("\nПлан перевозок методом северо-западного угла:");

i = j = 0;

for (i = 0; i < n; i++)

{

for (j = 0; j < m; j++)

{

Console.Write($"\t{C[i, j].Delivery}");

ResultCoast += (C[i, j].Value \* C[i, j].Delivery);

}

Console.Write($"\t{copieda[i]}");

Console.WriteLine();

}

for (int e = 0; e < m; e++)

{

Console.Write($"\t{copiedb[e]}");

}

Console.WriteLine("\nЗатраты на доставку: {0}\n", ResultCoast);

string[] potentialsA = new string[n];

string[] potentialsB = new string[m];

for (int q = 0; q < n; q++)

{

potentialsA[q] = "";

}

for (int w = 0; w < m; w++)

{

potentialsB[w] = "";

}

for (int w = 0; w < n; w++)

{

for (int e = 0; e < m; e++)

{

if (C[w, e].Delivery != 0)

{

if (potentialsA[w] == "")

{

if (w == 0 && e == 0)

{

potentialsA[w] = "0";

potentialsB[e] = (C[w, e].Value - int.Parse(potentialsA[w])).ToString();

}

else

{

potentialsA[w] = (C[w, e].Value - int.Parse(potentialsB[e])).ToString();

}

}

if (potentialsB[e] == "")

{

potentialsB[e] = (C[w, e].Value - int.Parse(potentialsA[w])).ToString();

}

}

}

}

Console.WriteLine("Опорный план:");

Console.Write($"\t\t");

for (int e = 0; e < m; e++)

{

Console.Write($"{potentialsB[e]}\t");

}

Console.WriteLine();

for (int w = 0; w < n; w++)

{

Console.Write($"\t{potentialsA[w]}");

for (int e = 0; e < m; e++)

{

Console.Write($"\t{C[w, e].Delivery}");

}

Console.Write($"\t{copieda[w]}");

Console.WriteLine();

}

Console.Write($"\t\t");

for (int e = 0; e < m; e++)

{

Console.Write($"{copiedb[e]}\t");

}

Console.WriteLine();

int[,] freecells = new int[n,m];

for (int w = 0; w < n; w++)

{

for (int e = 0; e < m; e++)

{

if (C[w, e].Delivery == 0)

{

freecells[w,e] = int.Parse(potentialsA[w]) + int.Parse(potentialsB[e]) - C[w, e].Value;

}

}

}

Console.WriteLine("\nОценка свободных клеток:");

Console.Write($"\t\t");

for (int e = 0; e < m; e++)

{

Console.Write($"{potentialsB[e]}\t");

}

Console.WriteLine();

for (int w = 0; w < n; w++)

{

Console.Write($"\t{potentialsA[w]}");

for (int e = 0; e < m; e++)

{

Console.Write($"\t{freecells[w,e]}");

}

Console.Write($"\t{copieda[w]}");

Console.WriteLine();

}

Console.Write($"\t\t");

for (int e = 0; e < m; e++)

{

Console.Write($"{copiedb[e]}\t");

}

Console.WriteLine();

foreach (int cell in freecells)

{

if (cell > max) max = cell;

}

Console.WriteLine($"\nНаибольшая оценка {max}");

}

}

}