Уральский федеральный университет имени первого Президента

России Б.Н.Ельцина

Институт радиоэлектроники и информационных технологий – радиотехнический факультет

Департамент информационных технологий и автоматики

**ОТЧЕТ**

по практическим занятиям

по дисциплине “Теория управления”

Преподаватель: доцент Осипова И. А.

Студент: Добрынин А. Д.

Группа: РИ-320950

Екатеринбург - 2025

**СОДЕРЖАНИЕ**

[**Практическое работа №1** 3](#_Toc195823195)

[**Практическое работа №2** 7](#_Toc195823196)

[**Практическая работа №3** 10](#_Toc195823197)

[**Практическое работа №4** 16](#_Toc195823198)

**Практическое работа №1**

Текущие дисциплины обучения по направлению Программная инженерия.

**Цель:** проанализировать все изученные дисциплины, распределить их по уровням, группам и выяснить какая сфера преобладает больше всего

**Задачи:**

* Создать папки для заметок по дням недели;
* Создать заметки для предметов;
* В заметках для предметов расписать конкретные задачи по каждому предмету;
* Выписать все изученные дисциплины из электронной зачетной книжки;
* Распределить все по папкам;
* Выписать дисциплины из прошлых семестров;
* Распределить дисциплины по группам (курс, семестр);
* Распределить дисциплины по уровням;
* Проанализировать результат.

**Теоретический аспект: Метод познания от общего к частному называется дедукция. Метод познания от частного к общему называется индукция.** В данной практической работе мы использовали оба метода, так как в начале мы шли от частного к общему, а потом от общего к частному. В начале мы расписывали текущие дисциплины и задачи к ним. Для эффективного управления задачами используем методику разбивки задач по неделям. Допустим: в воскресенье вечером нам нужно сделать какие-то определенные задачи по предметам к понедельнику и при этом свободном времени в понедельник у нас очень мало (примерно час, но этот час тратиться на дорогу, обед или ужин), так как помимо учебы, есть еще и работа. То есть задачи, которые нам нужно сделать к понедельнику, мы делаем в воскресенье.

Задачи записываем по мере их поступления и записываем их следующим образом:

* К какому дню недели нам нужно сделать задание по данному предмету;
* В какой день мы будем это делать.

При этом мы сверяемся с расписанием и проверяем на наличие конфликтов. Если подходит какой-то дедлайн, то соответственно в приоритете мы делаем этот предмет. Таким образом мы структурировали свои задачи.

Методологически мы опираемся на аналитический метод, что позволяет эффективно организовывать рабочий процесс.

Далее следует выписать все изученные дисциплины с прошлых семестров и распределить их по группам, тегам и так далее. И в конце, опираясь на тот же аналитический метод и метод дедукции, нужно выявить связи между предметами, так как в любом случае некоторые изучаемые дисциплины должны быть связаны между собой. Например: полученные знания на каком-нибудь курсе по программированию, моделированию или проектированию (например проектирование игр на движке Unity) можно применить на дисциплине «Проектный практикум» или, например если курс ведет компания-партнер, то можно попытаться попасть к ним на стажировку, тем самым закрыв производственную практику.

Используя все эти методы и подходы к обучению, нам нужно выяснить, какая сфера преобладает больше всего, и какая профессия нам подходит больше.

**Ход работы:** для начала мы создаем папки со всеми днями недели и далее добавляем в них заметки предметов. В оглавлении заметки мы пишем название предмета и далее описываем все задачи по этому предмету, которые нужно сделать к этому дню. В качестве вспомогательных материалов мы использовали следующие сервисы:

* Modeus;
* ЛК УрФУ;
* Teams;
* Telegram.

Данные сервисы помогли нам структурировать предметы в программе obsidian.

Далее мы открываем электронную зачетную книжку и выписываем все изученные нами дисциплины за все 3 года обучения. Потом мы распределяем их по папкам (курс, семестр) и дальше занимаемся созданием групп, по которым мы будем распределять эти дисциплины. В ходе анализа всех предметов мы выявили следующие группы:

* Базовые предметы;
* Данные и искусственный интеллект;
* Естественные науки;
* Компьютерные системы и инфраструктура;
* Математика и статистика;
* Практики;
* Программирование и технологии разработки;
  + Программирование на бизнес-ориентированных языках (подгруппа);
* Проектный практикум;
* Социальные и гуманитарные науки;
* Физическая культура.

Следующим шагом было поместить в эти группы ссылки на дисциплины, которые к этим группам относятся. После успешного распределения дисциплин по группам мы создаем теги «1Курс», «2Курс» и «3Курс» и присваиваем его всем этим группам. Далее методом анализа, мы выявляем связи между группами и дисциплинами. Оказалось, что среди изученных мною дисциплин есть много связанных между собой.

В заключении мы настраиваем фильтры графа, а именно: включаем чекбокс теги, включаем чекбокс объекты без связей, создаем новую группу по тегу «1Курс», «2Курс», «3Курс» и настраиваем отображение (порог исчезания текста, размер узла и толщина линии), а также настраиваем расстояние между узлами и силы притяжения, отталкивания и связей объектов. Для наглядности мы запускаем анимацию и смотрим, что у нас получилось. Далее вносим некоторые правки и корректировки, еще раз более точно настраиваем фильтры и в результате получаем следующее:

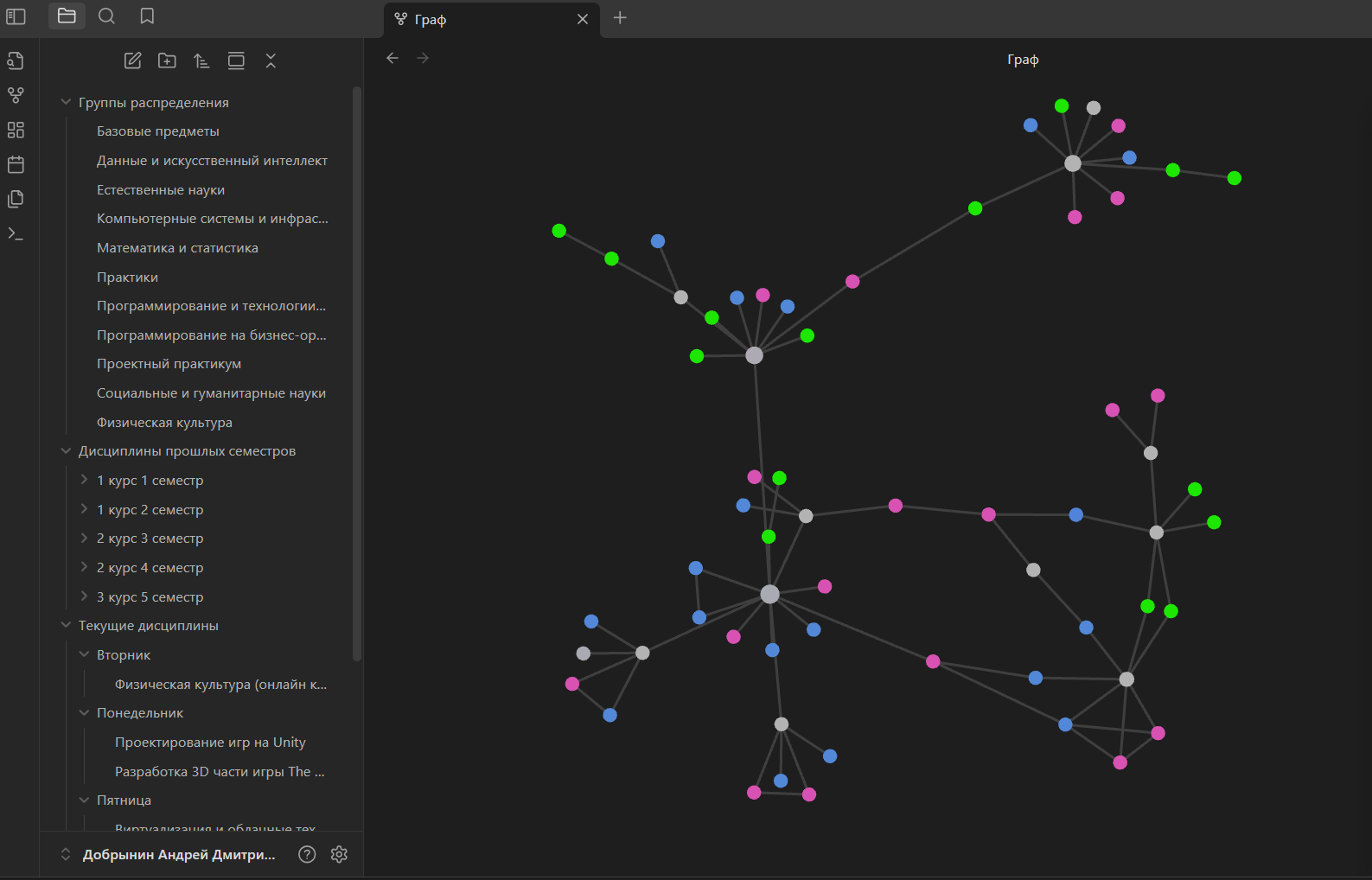


Рисунок 1 - результат работы

Теперь нам нужно проанализировать полученный результат. Как мы видим, на графе явно преобладают две сферы: разработка на бизнес-ориентированных языках и разработка игр. Как мы видим, данные сферы потенциально отличаются друг от друга, однако данные сферы, однако они связаны между собой и предоставляют выбор: в какой сфере больше развиваться? Возможно развиваться в этих сферах параллельно? Нужно ли что-то минимизировать для уменьшения нагрузки и для получения большей эффективности в другой сфере?

**Вывод:** в ходе работы, я изучил основную структуру программы obsidian и расписал в нем свои задачи по учебе, а также выявил связи изученных дисциплин между собой. Я считаю, что obsidian неплохой инструмент для данной задачи, так как я могу видеть все свои задачи на одном экране и грамотно структурировать их по предметам и неделям. Благодаря этому я могу рационально распределять нагрузку и повысить свою продуктивность в целом. Также, было полезно узнать, насколько хорошо я планировал свой учебный план и много ли предметов связаны между собой. Плагин «Граф» очень хорошо помог наглядно выстроить эти связи.

**Практическое работа №2**

**Цель:** изучить какое направление больше всего преобладает в моем учебном плане.

**Задача:** построение дерева кратчайшей длины (суммы весов входящих в него ребер), соединяющего N заданных вершин, без привлечения добавочных узлов.

**Теоретический аспект:** В данной работе рассматривается проблема построения минимального остовного дерева (МОД) для множества дисциплин учебного плана. МОД представляет собой связный подграф, содержащий все заданные вершины и имеющий минимальную суммарную длину рёбер. Применение данного метода позволяет выявить наиболее значимые дисциплины и их взаимосвязи, что способствует оптимизации образовательного процесса. В качестве методов мы будем использовать следующее:

Методы дедукции и индукции;

Аналитический метод;

Методы теории графов для построения минимального дерева.

**Ход работы:**

Для решения данной задачи мы будем рассматривать задачи Прима. Под задачей Прима понимается построение дерева кратчайшей длины (суммы весов входящих в него ребер, соединяющего N заданных вершин, без привлечения добавочных узлов. Для начала мы порешаем эти задачи для того, чтобы разобраться как правильно строить дерево кратчайшей длины, чтобы потом интерпретировать это на свои дисциплины. После изучения и решения данных задач, появилось представление, что нужно делать. Для начала мы сформировали граф дисциплин (Практическая работа №1), разделили их на сферы, определили связи между ними и назначили веса ребрам. Далее мы приступили к построению минимального остовного дерева, при этом мы применили алгоритм Прима для выделения наиболее значимых связей. Мы взяли 2 сферы для примера и сначала рассмотрели их отдельно. Далее мы объединили их в единую таблицу для комплексного анализа. В ходе анализа мы определили наиболее взаимосвязанные дисциплины, выявили преобладающие направления обучения и проанализировали наличие ключевых дисциплин, связывающих разные сферы. Теперь мы можем сформировать выводы и рекомендации:

Определение направлений, требующих углубленного изучения.

Оптимизация учебного плана для более логичной и эффективной последовательности освоения дисциплин.

Оценка возможности совмещения направлений для расширения профессиональных перспектив студента.

Применение данного подхода позволяет получить наглядное представление о структуре образовательной программы, выявить её сильные стороны и возможные пробелы, а также принять обоснованные решения о дальнейшем образовательном развитии.

**Результаты работы:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Предметы** | **Базы данных** | **Разработка 3D части игры The revenants** | **3D-моделирование** | **Проектирование игр на Unity** |
| **Базы данных** | ∞ | 5 | 6 | 3 |
| **Разработка 3D части игры The revenants** | 5 | **∞** | 2 | 2 |
| **3D-моделирование** | 6 | 2 | **∞** | 4 |
| **Проектирование игр на Unity** | 3 | 2 | 4 | **∞** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Предметы** | **Разработка на 1С** | **Производственная практика** | **Системная аналитика** | **Виртуализация** | **Векторный анализ** |
| **Разработка на 1С** | ∞ | 1 | 4 | 4 | 4 |
| **Производственная практика** | 1 | ∞ | 5 | 5 | 6 |
| **Системная аналитика** | 4 | 5 | ∞ | 4 | 4 |
| **Виртуализация** | 4 | 5 | 4 | ∞ | 3 |
| **Векторный анализ** | 4 | 6 | 4 | 3 | ∞ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Предметы** | **Разработка на 1С** | **Разработка 3D части игры The revenants** | **Базы данных** | **Программирование на JavaScript** | **Виртуализация** |
| **Разработка на 1С** | ∞ | 5 | 4 | 3 | 4 |
| **Разработка 3D части игры The revenants** | 5 | ∞ | 5 | 4 | 5 |
| **Базы данных** | 4 | 5 | ∞ | 3 | 4 |
| **Программирование на JavaScript** | 3 | 4 | 3 | ∞ | 3 |
| **Виртуализация** | 4 | 5 | 4 | 3 | ∞ |

Изображение выглядит как линия, диаграмма, астрономия

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2 - полученный граф

**Вывод:** Применение методов теории графов позволяет выявить основные направления в изучаемых дисциплинах, минимизируя количество промежуточных предметов и выявляя основные связи. Это может помочь в выборе будущей специализации и упрощении планирования обучения.

**Практическая работа №3**

**Цель:** изучить какое направление больше всего преобладает в моем учебном плане.

**Задача:** нахождение дерева с минимальным весом в мультиграфе.

**Теоретический аспект:** В этой работе мы продолжаем анализировать структуру моего учебного плана, используя методы, основанные на теории графов и статистическом анализе. Главная идея – на основе построения минимального остовного дерева понять, какие дисциплины и направления в учебном плане играют ключевую роль и как они связаны между собой.

**Используемые методы:**

Дедукция и индукция - мы начинаем с общих принципов теории графов (дедукция) и применяем их к конкретным данным моего учебного плана. Затем, анализируя полученные результаты (индукция), делаем выводы о взаимосвязях между дисциплинами.

Мы начинаем с общих принципов теории графов (дедукция) и применяем их к конкретным данным моего учебного плана. Затем, анализируя полученные результаты (индукция), делаем выводы о взаимосвязях между дисциплинами.

**Методы теории графов.**

Здесь я использую один основной алгоритм - **алгоритм Прима**. Он помогает начать с одной дисциплины и постепенно добавлять к «дереву» новые, выбирая ребра с наименьшим весом. Особенностью этого алгоритма является то, что он позволяет находить дерево с минимальным весом в любом мультиграфе. В данном алгоритме задаются веса соответствующих дуг мультиграфа неупорядоченным списком кодов вида: i, j, aij, uij, где i, j – номера вершин, aij – вес дуги, uij – совокупность кодов для всех дуг графа.

Сам алгоритм выглядит следующим образом:

**Шаг 1:** составляем для графа информацию в виде совокупности кодов, которые мы представили в теоретическом аспекте, массив <L>.

**Шаг 2:** просматриваем массив <L> и отмечаем значком «минус» те коды, в которых на месте i или j стоит номер, зафиксированный в ячейке d0 (в начальный момент это будет номер). Если рассматриваемый код из <L> ранее был уже отмечен «минусом» и должен быть вновь отмечен, то он из <L> вычеркивается и в дальнейшем уже просматриваться не будет.

**Шаг 3:** в отмеченных знаком «минус» кодах <L> выбираем тот, в котором значение aij, минимально и заносим этот код в массив <D>, после чего из <L> его вычеркиваем.

**Шаг 4:** проверяем количество занесенных в <D> кодов включая (0, к,0). Если количество их равно N, то дерево с минимальной длиной найдено – это есть совокупность всех кодов в <D>, не считая первого.

**Шаг 5:** в массиве <D> фиксируем последний занесенный в него код (i, j, aij). Находим вершину с номером i0 или j0, которая не встречалась в кодах.

**Ход Работы:** данную работу мы будем выполнять по этому алгоритму:

У нас есть следующие вершины:

1. Проектный практикум;
2. 3D-моделирование;
3. Проектирование на Unty;
4. Разработка на 1С;
5. Системная аналитика;
6. Виртуализация;
7. Векторный анализ;
8. Производственная практика;
9. Базы данных;
10. JavaScript;

Теперь приступим к самому алгоритму:

**Шаг 1:** для начала мы составим таблицу, в первом и во втором столбце которой будут предметы, которые соединены ребрами, а в третьем столбце мы пропишем веса этих ребер.

**Шаг 2:** теперь просматриваем нашу составленную таблицу и отмечаем знаком минус предмет «Проектный практикум». Так как это первый предмет в списке, мы начнем с него. Ищем все строчки с этим предметом и отмечаем их цифрой «1».

**Шаг 3:** далее ищем среди выделенных строчек, строчку с минимальным весом и выписываем эту строчку в таблицу 3 (в массив <D>) и вычеркиваем эту строчку. Далее повторяем те же действия с предметом «3D-моделирование».

**Шаг 4:** когда количество заполненных строк таблицы 3 будет равняться 11, то в таком случае переходим к шагу 5.

**Шаг 5:** в таблице 3 фиксируем последний занесенный в него код (**предмет**, **предмет**, **вес**). Находим вершину с номером **предмет**0, которая не встречалась в кодах.

**Таблица 1:**

|  |  |  |
| --- | --- | --- |
| **Предметы** | | **Веса** |
| Проектный практикум | 3D-моделирование | 5 |
| Проектирование игр на Unity | Проектный практикум | 4 |
| Производственная практика | Разработка на 1С | 5 |
| Базы данных | Системная аналитика | 5 |
| Базы данных | JavaScript | 2 |
| Проектный практикум | Разработка на 1С | 3 |
| Базы данных | Системная аналитика | 5 |
| JavaScript | Проектный практикум | 1 |
| 3D-моделирование | Проектирование игр на Unity | 4 |
| JavaScript | Виртуализация | 3 |
| Векторный анализ | Системная аналитика | 2 |
| Разработка на 1С | Системная аналитика | 3 |
| Виртуализация | Разработка на 1С | 3 |
| Системная аналитика | Векторный анализ | 2 |
| Проектный практикум | Производственная практика | 1 |
| Базы данных | Векторный анализ | 2 |
| Виртуализация | Проектный практикум | 1 |
| Базы данных | Виртуализация | 4 |
| 3D-моделирование | Производственная практика | 2 |

**Таблица 2:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Предметы** | | **Веса** |
| 1 2 | Проектный практикум | 3D-моделирование | 5 |
| 1 3 | Проектирование игр на Unity | Проектный практикум | 4 |
| 4 8 | Производственная практика | Разработка на 1С | 5 |
| 5 9 11 | Базы данных | Системная аналитика | 5 |
| 9 | Базы данных | JavaScript | 2 |
| 4 | Проектный практикум | Разработка на 1С | 3 |
| 5 9 11 | Базы данных | Системная аналитика | 5 |
| 1 10 | JavaScript | Проектный практикум | 1 |
| 2 3 | 3D-моделирование | Проектирование игр на Unity | 4 |
| 6 10 | JavaScript | Виртуализация | 3 |
| 5 7 | Векторный анализ | Системная аналитика | 2 |
| 4 5 11 | Разработка на 1С | Системная аналитика | 3 |
| 4 6 | Виртуализация | Разработка на 1С | 3 |
| 5 | Системная аналитика | Векторный анализ | 2 |
| 1 | Проектный практикум | Производственная практика | 1 |
| 7 9 | Базы данных | Векторный анализ | 2 |
| 6 | Виртуализация | Проектный практикум | 1 |
| 6 9 | Базы данных | Виртуализация | 4 |
| 2 | 3D-моделирование | Производственная практика | 2 |

**Таблица 3:**

|  |  |  |
| --- | --- | --- |
| **Предметы** | | **Веса** |
| Проектный практикум | Производственная практика | 1 |
| 3D-моделирование | Производственная практика | 2 |
| Проектирование игр на Unity | Проектный практикум | 4 |
| Проектный практикум | Разработка на 1С | 3 |
| Системная аналитика | Векторный анализ | 2 |
| Виртуализация | Проектный практикум | 1 |
| Векторный анализ | Системная аналитика | 2 |
| Базы данных | JavaScript | 2 |
| Производственная практика | Разработка на 1С | 5 |
| JavaScript | Проектный практикум | 1 |
| Разработка на 1С | Системная аналитика | 3 |

**Вывод:** в результате проделанной работы удалось глубже разобраться в структуре учебного плана. Применение методов теории графов позволило выявить ключевые дисциплины и направления, которые имеют наибольшее влияние на образовательную траекторию. Полученные результаты помогают не только увидеть преобладающие сферы (разработка игр, бизнес-ориентированное программирование и аналитика), но и дают возможность оптимизировать распределение ученой нагрузки, а точнее, помогают выявить на что обратить внимание больше, чтобы сделать обучение более эффективным и осознанным. Таким образом, проделанная работа стала хорошей основой для дальнейшего планирования и окончательного выбора специализации.

**Практическое работа №4**

**Цель:** изучить какое направление больше всего преобладает в моем учебном плане.

**Задача:**

**Теоретический аспект:**

**Ход Работы:**

Раннее начало работы = 5 + 2 + 5 = 12

Раннее окончание работы = 3 + 12 = 15

Позднее начало работы =

Позднее окончание работы =

Полный резерв времени =

Свободный резерв времени =

**Вывод:**