**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Кафедра ИМиИКТ им. В.В. Дика |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Информационные технологии и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | Очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по практической работе №2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | **Разработка и оформление технического задания** | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | **Технология разработки программного обеспечения** |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Ткаченко Ричард Николаевич |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | Группа – ДКИП 111 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | **Бунькин Виктор Иванович** |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г.**

**Цель работы:** приобретение навыков разработки технического задания на программный продукт, ознакомиться с правилами написания технического задания

Задание

Разработка технического задания

**Техническое задание на программный модуль "Решение комбинаторно-оптимизационных задач"**

1. **Введение**

1.1. Цель разработки модуля: Создание программного модуля для решения комбинаторно-оптимизационных задач, таких как задача коммивояжера, поиск кратчайшего пути и поиск минимального связывающего дерева.

1.2. Основные задачи модуля: - Разработка алгоритма поиска цикла минимальной длины (задача коммивояжера). - Реализация алгоритма поиска кратчайшего пути. - Создание алгоритма поиска минимального связывающего дерева.

2. **Требования к модулю**

2.1. Модуль должен быть разработан на языке программирования, предпочтительно Python.

2.2. Модуль должен предоставлять удобный интерфейс для интеграции с другими программами.

2.3. Предусмотреть возможность ввода графа (взвешенного или невзвешенного) для каждой из задач.

3. **Алгоритмы**

3.1. Разработать алгоритм поиска цикла минимальной длины: - Использовать метод полного перебора или эвристический метод для решения задачи коммивояжера.

3.2. Реализовать алгоритм поиска кратчайшего пути: - Можно использовать алгоритм Дейкстры или алгоритм Флойда-Уоршелла.

3.3. Разработать алгоритм поиска минимального связывающего дерева: - Использовать алгоритм Прима или алгоритм Крускала.

4. **Интерфейс пользователя**

4.1. Разработать текстовый или графический интерфейс для ввода данных (графа) и получения результата.

4.2. Обеспечить возможность визуализации полученных решений для наглядного анализа.

5. **Тестирование**

5.1. Предусмотреть тестовые сценарии для проверки корректности работы модуля.

5.2. Обеспечить возможность автоматического и ручного тестирования.

6. **Отчет**

6.1. Оформить подробный отчет, включающий: - Описание разработанных алгоритмов. - Примеры ввода и вывода. - Результаты тестирования. - Выводы и предложения по улучшению модуля.

7. **Нормативные требования**

7.1. Оформить программный модуль в соответствии с требованиями ГОСТ 19.106-78.

7.2. Предоставить документацию, описывающую использованные алгоритмы и структуры данных.

8. **Сроки**

8.1. Разработка и тестирование модуля должны быть завершены до [указать дату].

9. **Контроль и приемка**

9.1. Контроль качества и приемка осуществляются [указать ответственное лицо или команду].

10. **Финансирование**

10.1. Обеспечить финансовую поддержку разработки модуля в размере [указать сумму].

11. **Контактная информация**

11.1. Задачи и вопросы по проекту направлять [указать контактные данные].

Контрольные вопросы

1. Анализ требований. На этом этапе определяются потребности и ожидания пользователей, а также функциональные и технические требования к программному обеспечению.

Проектирование. На этом этапе создается архитектура программного обеспечения, которая определяет структуру, компоненты и взаимосвязи между ними.

Реализация. На этом этапе пишется код программного обеспечения в соответствии с разработанным проектом.

Тестирование. На этом этапе проводится тестирование программного обеспечения для выявления и устранения ошибок.

Развертывание. На этом этапе программное обеспечение устанавливается на целевые системы и настраивается для работы в производственной среде.

Поддержка. На этом этапе осуществляется поддержка программного обеспечения, которая включает в себя устранение ошибок, внесение улучшений и обновлений.

1. Постановка задачи и предпроектные исследования включают в себя следующие этапы:

* **Определение проблемы или возможности.** На этом этапе выявляется проблема, которую необходимо решить, или возможность, которую необходимо реализовать.
* **Сбор информации.** На этом этапе собирается информация, необходимая для понимания проблемы или возможности. Это может включать в себя проведение опросов, интервью, анализ данных и изучение существующих решений.
* **Анализ требований.** На этом этапе определяются потребности и ожидания пользователей, а также функциональные и технические требования к программному обеспечению.
* **Разработка концепции решения.** На этом этапе разрабатывается концепция решения, которая описывает основные принципы и подходы к решению проблемы или реализации возможности.
* **Оценка осуществимости.** На этом этапе оценивается осуществимость решения с технической, экономической и организационной точек зрения.
* **Разработка технического задания.** На этом этапе разрабатывается техническое задание, которое является документом, описывающим все требования к программному обеспечению.

Результатом постановки задачи и предпроектных исследований является четкое понимание проблемы или возможности, а также требований к программному обеспечению, необходимому для ее решения или реализации.

1. **Функциональные требования** определяют, что программное обеспечение должно делать, то есть какие функции оно должно выполнять. Они описывают поведение программного обеспечения с точки зрения пользователя. Функциональные требования могут быть:

* **Основные:** описывают основные функции программного обеспечения, без которых оно не может функционировать.
* **Дополнительные:** описывают дополнительные функции, которые повышают удобство использования программного обеспечения или расширяют его возможности.
* **Нефункциональные:** описывают характеристики программного обеспечения, которые не связаны с его функциональностью, например, производительность, надежность, безопасность и т.д.

**Эксплуатационные требования** определяют, как программное обеспечение должно работать в различных условиях. Они описывают поведение программного обеспечения с точки зрения системного администратора или пользователя. Эксплуатационные требования могут включать в себя:

* **Требования к производительности:** описывают, как быстро программное обеспечение должно выполнять свои функции.
* **Требования к надежности:** описывают, насколько устойчиво программное обеспечение должно работать в различных условиях.
* **Требования к безопасности:** описывают, как программное обеспечение должно защищать данные и ресурсы от несанкционированного доступа.
* **Требования к масштабируемости:** описывают, как программное обеспечение должно справляться с увеличением нагрузки или количества пользователей.
* **Требования к совместимости:** описывают, как программное обеспечение должно взаимодействовать с другими системами и приложениями.

1. **Правила разработки технического задания:**

**Ясность и полнота:** Техническое задание должно быть написано ясным и понятным языком, без двусмысленностей и неопределенностей. Оно должно содержать всю необходимую информацию о программном продукте, включая его цели, функции, требования к производительности, безопасности и т.д.

**Согласованность:** Техническое задание должно быть согласовано со всеми заинтересованными сторонами, включая заказчика, разработчиков, тестировщиков и системных администраторов. Это поможет избежать недопонимания и конфликтов на этапе разработки и внедрения программного продукта.

**Измеримость:** Техническое задание должно содержать измеримые требования, которые можно проверить и оценить. Это позволит контролировать ход разработки и убедиться, что программный продукт соответствует всем требованиям.

**Приоритезация:** Техническое задание должно содержать приоритеты для различных требований. Это поможет разработчикам сосредоточиться на наиболее важных функциях и требованиях, а также спланировать последовательность разработки.

**Управляемость:** Техническое задание должно быть управляемым, то есть оно должно быть достаточно подробным, чтобы обеспечить разработчикам ясное понимание требований, но в то же время не слишком громоздким и сложным.

**Реалистичность:** Техническое задание должно быть реалистичным, то есть оно должно учитывать возможности и ограничения разработчиков, а также сроки и бюджет проекта.

**Гибкость:** Техническое задание должно быть гибким, то есть оно должно допускать возможность внесения изменений в процессе разработки. Это позволит реагировать на меняющиеся потребности заказчика и учитывать новые технологии и тенденции.

1. **Титульный лист**. Содержит основную информацию о проекте, такую как название, цель, сроки и бюджет.

**Введение**. Описывает проблему, которую необходимо решить, и цели проекта.

**Требования**. Перечисляет функциональные и нефункциональные требования к системе.

**Дизайн**. Описывает архитектуру системы и ее основные компоненты.

**Реализация**. Описывает, как система будет реализована, включая выбор технологий и языков программирования.

**Тестирование**. Описывает, как система будет тестироваться, включая типы тестов и критерии приемки.

**Развертывание**. Описывает, как система будет развернута в производственной среде.

**Поддержка**. Описывает, как система будет поддерживаться после развертывания, включая обновления, исправления ошибок и обучение пользователей.