**ЗМІСТ**

[Введення 4](#_Toc417141561)

[1. Аналіз предметної області та класифікація сутностей для розв'язуваного завдання 5](#_Toc417141562)

[1.1. Аналіз предметної області 5](#_Toc417141563)

[1.2. Опис предметної області розв'язуваного завдання 5](#_Toc417141564)

[1.3. Функції предметної області, реалізовані завданням 5](#_Toc417141565)

[1.4. Класифікація сутностей 5](#_Toc417141566)

[2. Постановка завдання 6](#_Toc417141567)

[2.1. Організаційна сутність завдання 6](#_Toc417141568)

[2.2. Опис вихідної інформації завдання 6](#_Toc417141569)

[2.3. Формулювання завдання та вимог до системи 6](#_Toc417141570)

[3. Розробка інформаційного забезпечення завдання 7](#_Toc417141571)

[3.1. Опис інформаційної моделі системи 7](#_Toc417141572)

[3.2. Інформаційний аналіз вхідної інформації, необхідної для розв’язання завдання, і виділення інформаційних об'єктів предметної області 7](#_Toc417141573)

[3.3. Визначення зв'язків інформаційних об'єктів і побудова власної інформаційно-логічної моделі 7](#_Toc417141574)

[3.4. Визначення й обґрунтування логічної структури бази даних 7](#_Toc417141575)

[3.5. Розробка вихідних даних контрольного приклада 7](#_Toc417141576)

[3.6. Створення бази даних й її наповнення 7](#_Toc417141577)

[3.7. Опис технології уведення в базу даних вхідної інформації завдання 8](#_Toc417141578)

[4. Опис технології й алгоритмів рішення завдання та їх програмна реалізація 9](#_Toc417141579)

[4.1. Опис блок-схем алгоритмів роботи 9](#_Toc417141580)

[4.2. Узагальнений алгоритм рішення завдання і його декомпозиція на модулі 9](#_Toc417141581)

[4.3. Алгоритми реалізації окремих модулів завдання й результат їх функціонування 9](#_Toc417141582)

[5. Розробка інтерфейсу користувача 10](#_Toc417141583)

[5.1. Опис структури технологічного процесу обробки даних для рішення завдання. Опис програмних модулів. Опис роботи програми на контрольних прикладах (наприклад, на великому текстовому файлі), тимчасові характеристики програми на простих типах для одного з методів: сортування, балансування, пошук у неупорядкованій послідовності 10](#_Toc417141584)

[5.2. Розробка та реалізація прикладної програми 10](#_Toc417141585)

[5.4. Розробка інструкції користувача. Опис керівництва користувача 10](#_Toc417141586)

[5.5. Розподіл основного матеріалу курсового проекту по главах визначається автором курсового проекту 10](#_Toc417141587)

[Висновок 11](#_Toc417141588)

[Список використаних джерел 12](#_Toc417141589)

[Додатки 13](#_Toc417141590)

Введення

Призначенням даного проекту являється дослідження різних методів для пошуку екстремумів, максимуму та мінімуму функцій. Зараз все більше розвиваються комп′ютерні технології і стають важливими проблеми оптимізації та мінімізації математичних методів. Проблему оптимізації задач нелінійного програмування я вважаю актуальною на даний момент, тому обрала тему, яка розглядає одну з цих задач. Нелінійне програмування використовується для задач, які мають певні обмеження або цільова функція не є лінійною.

Введення є обов'язковим розділом пояснювальної записки до курсового проекту. У ньому повинна бути обґрунтована актуальність проблеми, розглянутої в проекті, оцінка сучасного стану розв'язуваної проблеми, новизна теми, огляд існуючих теоретичних концепцій, методів та розв’язання. З**вичайно** описуються переваги використання об'єкто-зорієнтованого програмування, мета, призначення та завдання, розв'язувані за допомогою створеної програми. Формулюються також власна точка зору по розглянутих питаннях, мета та завдання, які вирішуються в даному курсовому проекті.

Важливою принциповою вимогою вступної частини проекту є обґрунтування вибору об'єкта та предмета дослідження.

Об'єкт дослідження - це процес, реальний об'єкт або наукове явище, що породжує проблемну ситуацію та обране для вивчення (завод, цех, відділ, процес, система, явища й т.д.). Предмет - це те, що перебуває в границях об'єкта.

При завершенні вступної частини варто привести розгорнуте формулювання теми курсового проекту з точною вказівкою об'єкта проектування. При необхідності дається історичний екскурс, окреслюється коло проблем, що бідують у вивченні, визначається напрямок роботи.

Введення нумерується. Загальний обсяг не більше 5 сторінок**.**

1. ТЕХНІЧНЕ ЗАВДАННЯ

1.1. Підстави для розробки

Розробка програмного забезпечення виконується на підставі рішення засідання кафедри ПЗКС про затвердження тем дипломних та курсових робіт.

1.2. Призначення розробки

Призначенням даної розробки є створення програми для пошуку екстремумів, максимуму та мінімуму функції.

1.3. Аналіз вимог до програмного забезпечення

1.3.1. Функціональні вимоги

До програмного забезпечення висуваються наступні вимоги:

* Аналіз рядку, введеного користувачем
* Пошук максимуму функції
* Пошук мінімуму функції

1.3.2. Вимоги до складу та параметрів технічних засобів

Для нормальної роботи розробленого програмного продукту необхідний персональний комп’ютер з наступними мінімальними характеристиками:

|  |  |
| --- | --- |
| Параметр | Значення |
| Процесор | Intel Core 2 Duo 1.8 Ghz |
| Пам’ять | 1 GB RAM |
| Жорсткий диск | 40 GB |

1.3.3. Вимоги до інтерфейсу

Програма повинна мати інтуїтивно зрозумілий, зручний та візуально приємний інтерфейс.

Програмне забезпечення повинно бути оформлене у вигляді стандартного вікна операційної системи Windows.

Програма повинна складатися з:

* Системи введення символьного рядку
* Системи пошуку максимуму заданої функції
* Системи пошуку мінімуму функції

1.3.4. Вимоги до інформаційної та програмної сумісності

Необхідною вимогою до програмного забезпечення є встановлена операційна система класу Windows XP/7. Окремих вимог не висувається.

1.3.5. Вимоги до тестування програмного забезпечення

Для тестування програмного забезпечення необхідно виконати наступні дії:

* запустити програму;
* перейти до введення функції однієї змінної у символьний рядок;
* ввести межі та точність пошуку;
* вибрати пошук максимуму функції та натиснути кнопку результат;
* вибрати пошук мінімуму функції та натиснути кнопку результат;
* переглянути результати пошуку;
* повторити попередні дії з іншою функцією;
* закрити програму;

Після успішного виконання цих дій тестування програми може вважатися завершеним.

1.4. Вимоги до програмної документації

1.4.1. Склад супроводжувальної документації

Програмне забезпечення постачається разом із супроводжувальною документацією, в склад якої входить:

* технічне завдання;
* опис та обґрунтування вибраної архітектури;
* функціональна специфіка;
* технічна специфікація;
* опис програми;
* програма та методика випробувань;

1.4.2. Вимоги до супроводжувальної документації

Текст супроводжувальної документації набирається на комп’ютері шрифтом Times New Roman 14 пт. через 1,5 інтервал. Розміщення – книжкове.

Поля зліва – не менше 2,5 см., решта – не менше 1,5 см. Вирівнювання – по ширині. Відступ першого рядка абзацу – 1,27 см. Нумерація сторінок – наскрізна в усій роботі. Сторінки нумеруються по центру верхнього контитулу. На титульному аркуші та листі затвердження нумерація сторінок не ставиться, але вони враховуються в загальній кількості сторінок.

1.5. Стадії та етапи розробки

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Назва етапів кваліфікаційної розробки | Термін виконання етапів розробки | Примітка |
| 1 | Одержання технічного завдання | 01.12.2013 |  |
| 2 | Аналіз літератури | 15.01.2014р. |  |
| 3 | Розробка інтерфейсу | 20.02.2014р. |  |
| 4 | Реалізація програми | 02.03.2014р. |  |
| 5 | Тестування та налагодження | 05.04.2014р. |  |
| 6 | Оформлення програмної документації | 14.04.2014р. |  |
| 7 | Представлення готової роботи | 17.04.2014р. |  |
| 8 | Захист роботи | Згідно розкладу |  |

1.6. Порядок контролю і приймання

Програмне забезпечення повинне бути випробуване та протестоване на наявність помилок.

Порядок контролю і прийому проекту полягає в наступному:

* перевірка правильного функціонування програми керівником проекту;
* перевірка відповідності стандартам оформлення документації;
* перевірка правильного оформлення програмної документації керівником проекту;
* попередній захист курсової роботи;

2. АРХІТЕКТУРА, ФУНКЦІОНАЛЬНІ ТА ТЕХНІЧНІ ПОКАЗНИКИ

2.1. Призначення та область застосування

Призначення даної програми є дослідження різноманітних функції на екстремуми, максимум та мінімум.

2.2. Опис та обґрунтування обраної архітектури

Для даного проекту була вибрана об’єктно-зорієнтована архітектура, оскільки вона є найпоширенішою та найкраще підходить для розробки програм складного рівня. Були проаналізовані декілька методів пошуку максимуму та мінімуму функції і обрано ті, якій найбільш підходять даній задачі.

Роботу розробленої програми можна показати наступною блок-схемою, що показана на рис. 1.

**2.3.** Функціональна специфікація

2.3.1. Опис функціональних можливостей

Розроблена програма надає можливість реалізовувати пошук максимуму та мінімуму заданої в рядок функції. Її функціонал дозволяє користувачеві вводити символьний рядок з форми через кнопки, задавати межі знаходження функції та точність обчислень. Також програма надає можливість аналізувати результат пошуку двома методами: методом золотого січення і методом пошуку похідних.

2.**3.2.** Опис інтерфейсу користувача

**2.4.** Технічна специфікація

**2.4.1.** Опис і обґрунтування вхідних і вихідних даних

Вхідними даними для розробленого продукту є символьний рядок заданий через кнопки на формі, а також введені з клавіатури дані в поля про межі заданої функції і точність розбиття. Крім цього передбачений вивід про помилки введення. А також при натисненні кнопки результату буде виведена інформація про результат пошуку двома способами.

**2.4.2.** Опис модулів

Робота даного програмного забезпечення реалізується в наступних модулях:

* Main – реалізовано головну форму програми, за допомогою якої користувач взаємодіє з програмою;
* MathParser – клас, в якому реалізовано аналіз символьного рядка для обрахунку результату методом рекурсивного спуску;
* ParserResult – клас, де зберігається результат дій з парсером, а саме частина рядка, яку вже проаналізували і решту рядку;
* Extreme – клас, в якому реалізовано алгоритм пошуку похідної функції в точці, а також аналіз значень функції, де похідна дорівнює нулю і пошук максимуму та мінімуму;
* GoldenSection – клас, в якому реалізовано алгоритм пошуку максимуму та мінімуму функції методом золотого січення;
* MaxMinResult – клас, в якому зберігаються результати пошуку максимального та мінімального значення функцій;
* MyConvert – клас, в якому реалізовано переведення типів, таких як string, System:: String^ та double;
* MyExceptions – клас, в якому реалізовано виключні ситуації;

3. КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Опис і обґрунтування обраних програмних засобів

Розробка програмного забезпечення здійснена в середовищі Visual Studio 2013 C++. Дане інтегроване середовище розробки – зручний засіб розробки віконних програм для операційної системи Windows XP/7. В ньому поєднані потужні можливості мови програмування С++ та величезний набір візуальних компонентів, які, в свою чергу, дозволяють розробнику легко швидко та якісно створювати програмні продукти з візуальним інтерфейсом. Також, візуальна бібліотека містить елементи, що спрощують діалогову взаємодію користувача та додатку (діалогові вікна, відкриття та збереження файлів, налаштування параметрів кольорів, шрифту та іншого…)

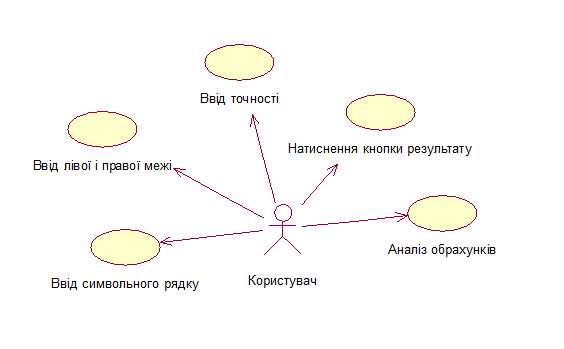
3.2. Опис програми

3.2.1. Функціональні можливості

В створеному програмному забезпеченні реалізовано наступні функції:

* Аналіз символьного рядку та його обрахунок;
* Пошук максимуму/мінімуму функції шляхом методу золотого січення;
* Пошук максимуму/мінімуму функції шляхом пошуку екстремумів функції;

3.2.2. Опис логічної структури

Логічну структуру даного програмного забезпечення зручно подати у вигляді наступної блок-схеми рис. 6.

3.2.3. Використані технічні засоби

Розробка програмного продукту здійснювалася на персональному комп’ютері з наступною конфігурацією:

* Процесор – Intel Pentium CPU B960 @ 2,2 GHz 2,2 GHz;
* ОЗП – 6 GB;
* Відеоадаптер – Nvidia Geforce GT 630M 1 GB;
* Жорсткий диск – 500 GB;
* DVD-RW – PLDS DS8A8SH ATA Device

Даний комп’ютер працює під управлінням ОС Windows 7.

4. ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

4.1. Об’єкт випробувань

Об’єктом випробувань є створене в процесі виконання курсової роботи програмне забезпечення. Для перевірки коректності роботи програми було проведено ряд тестів.

4.2. Використані технічні засоби

При проведенні тестування використовувалися технічні засоби, описані в підрозділі 3.3.1 «Використані технічні засоби». Під час проведення тестів використовувалася система ручного тестування.

4.3. Порядок та методика випробувань

Порядок проведення випробувань програмного забезпечення:

* запустити програму;
* перейти до введення функції однієї змінної у символьний рядок;
* ввести межі та точність пошуку;
* вибрати пошук максимуму функції та натиснути кнопку результат;
* вибрати пошук мінімуму функції та натиснути кнопку результат;
* переглянути результати пошуку;
* повторити попередні дії з іншою функцією;
* закрити програму;

4.4. Результати випробувань

Випробування програмного продукту проводилось шляхом тестування різних введених функцій. При проведенні випробувань серйозних недоліків не виявлено.

Висновок

Список використаних джерел

Додатки

Методи розв′язування задач нелінійного програмування поділяються на декілька видів. Ці види представлені на рис. 1.



Рис. 1. Види задач нелінійного програмування

Так як задача проекту полягає в пошуку екстремумів, а також максимуму та мінімуму функції, то було обрано два методи, якими буде розв′язана поставлена задача. Перший метод – метод золотого січення реалізовує лише пошук максимуму та мінімуму, а другий метод – метод пошуку похідних та екстремумів, а потім визначення максимуму та мінімуму.