МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Київський національний університет імені Тараса Шевченка

Механіко-математичний факультет

Кафедра алгебри та комп’ютерної математики

**Курсовий проект**

Програмування

(назва дисципліни)

на тему: «Клієнт-сервер для знаходження коренів

поліному довільного порядку»

Виконав: студент 3 курсу групи № 2

Спеціальності “Комп’ютерна математика”

(шифр і назва напряму підготовки (спеціальності))

Лісний А. І.

(прізвище й ініціали студента)

Керівник:

доцент, к.т.н., Бородін В.А.

(посада, науковий ступінь, прізвище й ініціали)

Національна шкала: \_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії:

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище й ініціали)

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище й ініціали)

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище й ініціали)

**ЗМІСТ**

Вступ……………………………………………………………………… 2

Теоретичні відомості …………………… ……………………………… 3

Огляд алгоритмів………………………………………………………… 6

Клієнт-серверна архітектура……….…………………………………… 11

Висновки………………………………………………………………… 12

Список використаної літератури……………………………………… 13

**ВСТУП**

Розроблена клієнт-серверна програма призначена для багатопоточної обробки клієнтів: сервер розв’язує математичну задачу, клієнт отримує розв’язки.

Математична задача полягає в знаходженні коренів поліному довільного порядку (n=1,2,3,4, n>5). Для знаходження коренів використовується алгоритм Лобачевського-Греффе.

Спілкування клієнта і сервера проходить по протоколу HTTP 1.0. Користувач за допомогою GET та POST вводить коефіцієнти поліному, сервер відповідає “202” або “404” з можливістю зберегти корені в форматі JSON.

**Теоретичні відомості**

Метод Лобачевського-Греффе

Нехай ми маємо рівняння вигляду:

(1)

для якого відомо, що вони дійсні і задовольняють умові:

(2)

Використовуючи формули Вієта маємо зв'язок між коренями і коефіцієнтами рівняння:

(3)

З першого рівняння винесемо x1:

За умовою (2) ми знаємо, що x1 за модулем більше решти коренів за абсолютною величиною, тому ми можемо знехтувати усіма відношеннями, які містяться в дужках. Отримаємо:

Аналогічним чином, з другого рівняння рівностей (3), винесемо за дужки x1x2:

Вважаючи, що відношеннями, які стоять в дужках, можна знехтувати по відношенню з одиницею, перепишемо формулу у наступному вигляді:

Продовжуючи ці міркування далі, отримаємо остаточну розрахункову формулу метода Лобачевського-Греффе, з допомогою якої можемо знайти наближені значення всіх коренів рівняння (1):

Слід зазначити, що такий підхід для знаходження коренів рівняння можливий лише в тому випадку, коли виконується умова (2).

Лобачевський М.І. запропонував спосіб отримання з даного рівняння (1) нового рівняння, корені якого рівні квадратам вихідного рівняння. Виконавши даний процес (квадрування) достатню кількість разів ми отримаємо нове рівняння, корені якого задовольняють умові (2).

Таким чином, ми зможемо знайти корені нового рівняння, а з них і корені початкового.

Запишемо рівняння (1) у вигляді:

Рівняння, корені якого протилежні за знаком кореням рівняння (1), матиме вигляд:

Перемножуючи ці два рівняння отримаємо:

Якщо покласти z=-x2, ми отримаємо нове рівняння відносно z, корені якого дорівнюють:

Звідси, для отримання необхідного нам рівняння, нам треба перемножити рівняння (1) і рівняння (1) с заміною x на -x, замінивши -x2 на z, отримаємо:

Коефіцієнт a\* при zn-k в цьому рівнянні виходить наступним чином:

Від квадрату ak віднімається подвоєний добуток двох сусідніх із ним симетрично розташованих коефіцієнтів, додається добуток наступних за ними симетрично розташованих коефіцієнтів і т. д., доки не прийдемо до a0 або an.

**Огляд алгоритмів**

Клієнт-серверна програма реалізована у вигляді двох частин: сервера і клієнта(client\_part, server\_part відповідно). Для реалізації JSON використовується бібліотека json.

**server\_part**

**Server.java:**

Основний клас сервера, що створює сокет сервера та в нескінченному циклі приймає нового клієнта та створює новий потік (thread) і “відправляє” клієнта на обробку до handler.

Для запуску серверу треба запускати саме цей клас, у разі потреби вказавши/змінивши порт у параметрі port.



**Handler.java:**

Клас для обробки клієнтів. Вхідними параметрами конструктору класу є сокет серверу і сокет клієнту (ServerSocket, Socket відповідно). В класі реалізовано інтерфейс Runable та оголошено метод run, що дозволяє зробити з об’єкта потік.

В полях класу зберігається інформація про типи даних, які може приймати сервер та клієнт, а також сокети сервера та клієнта.

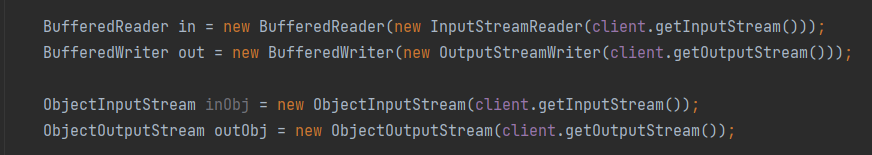
Для обміну даними використовуються потоки введення та виведення.

BufferedReader in – читає текст із потоку введення символів.

BufferedReader out – записує текст до потоку виведення символів.

ObjectOutputStream inObj – читає об’єкт з потоку.

ObjectOutputStream outObj - записує в потік окремий об'єкт.

****

Handler отримує команди у BufferedReader in і обробляє використовуючи метод readHttp в Functions.

Для формування http повідомлень використовується клас ServerHttpManager.

**Functions.java:**

Клас з допоміжними функціями:

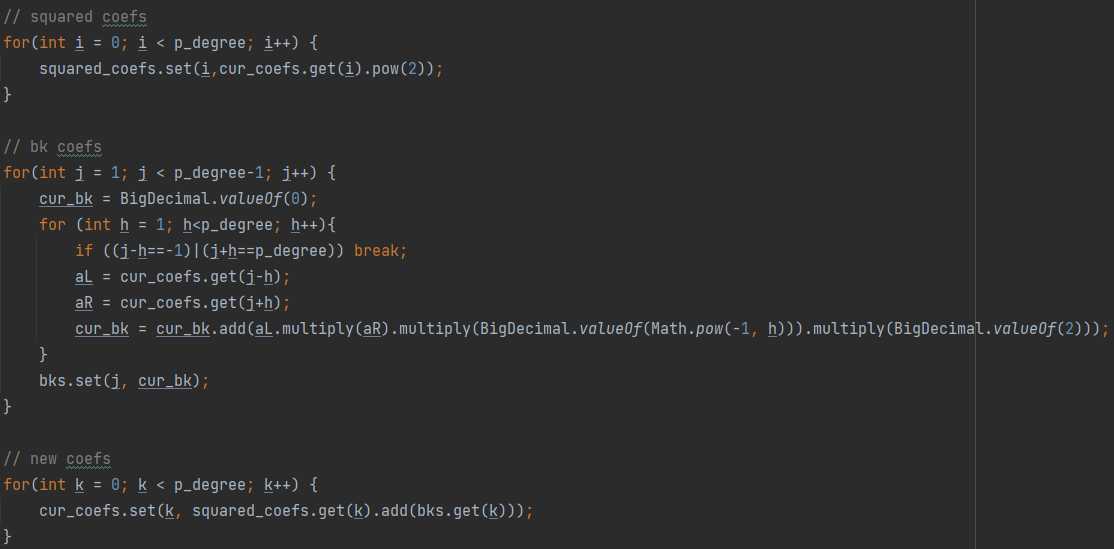
readHttp – приймає рядок http і повертає словник хедерів і їх параметрів.

enterPolinom – метод для введення поліному.

prt – метод для виведення списку об’єктів у рядок.

Graefee – метод Лобачевського-Греффе. Метод приймає список BigDecimal. BigDecimal використовуються через великі коефіцієнти після квадрування.

В основному циклі проходить процес зведення коефіцієнтів у квадрат і додавання симетричних часток (bk coefs).



Після проведення 10 ітерацій ми обчислюємо корені та додаємо в масив prob\_roots. Для кожного кореня тестуємо -x та x, підставляючи в початкове рівняння і видаляємо їх зі списку prob\_complex.

Якщо список prob\_complex не порожній – ми маємо комплексні корені. В списку залишаться модулі попарно спряжених комплексних чисел. Якщо таких чисел 2 або 4 використовуємо алгоритм знаходження цих чисел, інакше алгоритм поверне лише модулі комплексних чисел.

Алгоритм повертає список рядків коренів

**Grafee\_test.Java:**

Клас для тестування методу Лобачевського-Греффе.

**ServertHttpManager.java:**

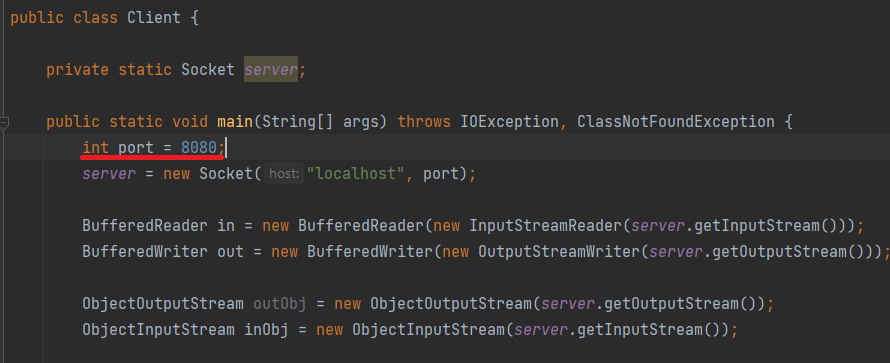
Конструктор приймає на введення сокет сервера. Методи OK та NotFound повертають відповіді 200 та 404 відповідно.

**client\_part**

**Client.java:**

Для обміну даними використовуються потоки введення та виведення ідентичні до Server.java.

Для запуску клієнта треба запускати саме цей клас, у разі потреби вказавши/змінивши порт у параметрі port.



**ClientHttpManager.java:**

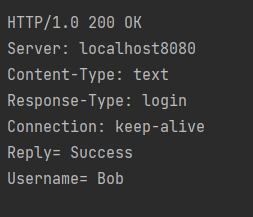
Конструктор приймає на введення сокет сервера. Методи GET та POST повертають рядки http запитів.

**Клієнт-серверна архітектура**

Програма складається з двох програм. Клієнтська програма, що працює на стороні клієнта, та серверна програма, що працює на стороні сервера. Багатопочність сервера забезпечує підтримку множини клієнтів. Спілкування сервер-клієнт проходить по протоколу http. Командою POST клієнт передає інформацію про типи, що підтримуються клієнтом і типи, що клієнт надає. Командою GET клієнт відправляє запит на корені в форматі JSON.

****

Cервер відповідає 200 або 404 якщо виникла помилка.

****

**Висновки**

В процесі виконання курсової роботи я познайомився клієнт-серверною архітектурою та багатопочністю. Багатонитковість дозволяє серверу обробляти декількох клієнтів одночасно, а не очікувати обробки кожного клієнта.

Реалізована програма знаходить корені з високою точністю після коми завдяки Bigdecimal. Цей клас використовується для обчислень із вкрай високими вимогами до точності або розміру даних. В програмі при потребі можна збільшити кількість виведених знаків після коми з 5 до потрібної кількості.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. <https://habr.com/ru/post/330676/>

2.https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4\_%D0%9B%D0%BE%D0%B1%D0%B0%D1%87%D0%B5%D0%B2%D1%81%D1%8C%D0%BA%D0%BE%D0%B3%D0%BE\_%E2%80%94\_%D0%93%D1%80%D0%B5%D1%84%D1%84%D0%B5

3. Березин И.С., Жидков Н.П. - Методы вычислений. Том 2

4. https://javarush.ru/groups/posts/654-klassih-socket-i-serversocket-ili-allo-server-tih-menja-slihshishjh