



โครงการคอมพิวเตอร์

วิชา Object oriented programming

จัดทำโดย

6404062610090 นันทินี แสงโชคพาหะ

เสนอ

อาจารย์สถิตย์ ประสัมพันธ์

ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

คณะวิทยาศาสตร์ประยุกต์ สาขาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ปีการศึกษา 2564

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ปัจจุบันการพัฒนาด้านเทคโนโลยีคอมพิวเตอร์และการสื่อสารได้นำมาใช้เป็นเครื่องมือ ช่วยในการศึกษา การเรียนรู้ให้มีประสิทธิภาพและสะดวกสบายมากยิ่งขึ้น เราเล็งเห็นถึงความสำคัญในจุดนี้ จึงมีความคิดที่จะพัฒนาเว็บไซต์คำนวณตัวเลขรวมหัวข้อเรื่องต่างๆไว้ เพื่อใช้คำนวณตัวเลขที่มีกรคิด ค่าคำนวณที่เยอะ มาแสดงค่าคำตอบและวิธีทำต่างๆได้ โดยการใช้การเขียนโค้ดที่ได้ศึกษามาให้เกิดประโยชน์

1.2 ประเภทโครงการ

เป็นโครงการประเภทสาระการเรียนรู้

1.3 ประโยชน์

1.3.1 เพื่อให้ผู้ที่สนใจเกี่ยวกับการคำนวณนำไปประยุกต์ใช้ต่อ

1.3.2 ช่วยตรวจสอบคำตอบของฟังก์ชันที่ต้องการได้อย่างรวดเร็ว

1.3.3 สามารถใช้เป็นสื่อประกอบการเรียนการสอนให้เพิ่มความน่าสนใจขึ้น

1.3.4 ช่วยผู้ที่ต้องการมาพัฒนาโปรแกรม หรือ ผู้ที่เข้ามาศึกษามาใช้ สามารถพัฒนาต่อได้ใน

อนาคต

บทที่ 2

ส่วนการพัฒนา

2.1 รูปแบบการพัฒนา

2.2.1 เพิ่มฟังก์ชันคำนวณ

2.2.2 เพิ่มการตกแต่งให้มีความสวยงามมากขึ้น + เพิ่มหน้าต่างตัวเลือก

2.2.3 เพิ่มกราฟ เมตริก ปุ่ม ช่องกรอกค่า

2.2 อธิบายส่วนของโปรแกรม

```
import React,{ useState } from 'react' // ตัวนี้เป็นฟังก์ชัน component
import {Chart, Series} from 'devextreme-react/chart';
import { render } from '@testing-library/react';

const Parser = require('expr-eval').Parser; // แปลง string to ???

const Bisection = () => {
  const [Keepvalue,setKeepvalue] = useState ([])
  const [CheckUnValid,setCheckUnValid] = useState (false)
  const [ans,setans] = useState ()
  const [err,seterr] = useState ()
  var arr = []
  const arrG = [{value1: 'error', value2: 'iteration'}];
```

ตัวแปร Parser มีไว้เพื่อรับค่าแล้วแปลงค่าที่รับเข้ามาจาก String เป็น ตัวเลข

ตัวแปร Bisection ที่เป็น arrow function นั้นเหมือนกับ class ใน OOP

และมีการประกาศ parameter ซึ่งในที่นี้ใช้ useState ในการเซตค่าของตัวแปร

```
const Bisection_solve = (XL,XR>Error,Functions) => {
  arr.splice(0,arr.length)
  const parser = new Parser();

  function fx(x) {
    let func = parser.parse(Functions)
    return func.evaluate({ x: (x) }) //ส่งค่ากลับไปใช้
  }

  var xl = parseFloat(XL); // String to float
  var xr = parseFloat(XR);
  var xm,xold;
  var Errors = 10000000; //รอบสูงสุด
  var InputError = parseFloat(Error);
  var TempArray = [];
  var i = 0;
```

สร้าง function Bisection_solve ที่รับค่า XL(ขอบซ้าย), XR(ขอบขวา), Error(ค่า error ที่ยอมรับ), Functions เข้ามาเพื่อใช้ในการคำนวณ

ตรง function fx(x) คือ แปลง Functions ที่รับเข้ามาจาก String เป็น ตัวเลข(ทำให้สามารถคำนวณได้)

xl, xr แปลงค่าที่รับเข้ามาจาก String เป็น float

```
if(xl!=null && xr!=null && Functions!=null && InputError!=null)
{
  while(Errors > InputError)
  {
    xm=((xr+xl)/2);
    if(fx(xm)*fx(xr)<0)
    {
      xold=xl;
      xl=xm;
    }
    if(fx(xm)*fx(xr)>0)
    {
      xold=xr;
      xr=xm;
    }
    i++;
    Errors = Math.abs((xm-xold)/xm)*100;
    arr.push({xl : parseFloat(xl) , xm : parseFloat(xm) , xr : parseFloat(xr) , error : parseFloat(Errors) , iteration : parseFloat(i)})
    let Count = [xl.toFixed(6),xr.toFixed(6),xm.toFixed(6),Errors.toFixed(6)]; // item [0],[1],[2],[3]
    TempArray.push(Count);
  }
  setans(xm.toFixed(6));
  seterr(Errors.toFixed(6));
}
setKeepvalue(TempArray);
```

เข้าเงื่อนไขเพื่อคำนวณหาคำตอบ โดยจะเช็คค่า xl, xr, Functions และ InputError ไม่ใช่ค่าว่าง แล้วให้เข้า loop while เมื่อค่า Errors ที่คำนวณได้ มากกว่าค่า InputError แล้วจะนำ xl(ขอบซ้าย)มาบวกกับxr(ขอบขวา) หลังจากนั้นหารด้วย 2 จะได้ค่ากลาง แล้วนำค่าที่ได้ไปแทนลงในสมการแล้วจะได้ค่า ทำอย่างนี้ไปจนกว่าค่าที่ได้ จะน้อยการ Error ที่ต้องการ

```

render(
  <div className="graph" id='graph'>
    <Chart palette="Violet" dataSource={arr}>
      {arrG.map((item) =><Series
        argumentField = {item.value2}
        valueField = {item.value1}
        name = "Error"
        type = "line"
        color = "#000d6c"
      />)}
    </Chart>
  </div>
)
}

```

ในส่วนนี้คือการแสดงกราฟ บนหน้า website

```

const InputNumber = (e) =>
{
  e.preventDefault()
  let xl = e.target.XL.value
  let xr = e.target.XR.value
  let errorer = e.target.Error.value
  let fu = e.target.Function.value
  if(xl && xr && errorer && fu)
  {
    setCheckUnValid(false); // ถ้ามี
    Bisection_solve(xl,xr,errorer,fu);
  }
  else
  {
    setCheckUnValid(true);
  }
}

```

Function นี้คือ function ที่เอาไว้เช็คค่ารับค่าครบทุกค่าหรือไม่ ถ้าไม่จะไม่สามารถคำนวณได้ แต่ถ้าใส่ค่าครบทุกค่าจะเรียกใช้ Bisection_solve เพื่อหาคำตอบ

```

return(
  <div className='box'>
    <div className="title">
      <h1>Bisection Method</h1>
    </div>
    <div className="content">
      <div className="input-box">
        <form onSubmit={InputNumber} className="form">
          <div className="in">
            <div className="inbox">
              <label> &nbsp;XL :&nbsp; </label>
              <input name='XL' placeholder='Starting XL' />

              <label> &nbsp;XR :&nbsp; </label>
              <input name='XR' placeholder='Starting XR' />
            </div>
            <p></p>
            <div className="inbox">
              <label className='text'> &nbsp;Error:&nbsp; </label>
              <input name='Error' placeholder='Epsilon' />

              <label className='text'> &nbsp;Function :&nbsp; </label>
              <input name='Function' placeholder='Input function here' />
            </div>
            <p></p>
          </div>
          <div className="inbox">
            <div className="inbox-btn">
              &nbsp;<button id='btn'>Calculate</button>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>

```

ส่วนที่แสดงบนหน้า website โดย การรับค่าจะอยู่ที่ div className="input-box" เป็นต้นไป ตรง form
onSubmit={InputNumber} คือเมื่อมีการกดปุ่มคำนวณจะเรียกใช้ InputNumber

```

<div className="showans">
  {CheckUnValid && <h4 id="ans">Input XL,XR,Error and Function first!</h4>}
  {!CheckUnValid && <h4 id="ans">The value of root is : {ans}</h4>}
  {!CheckUnValid && <h4 id="err">Error is : {err}</h4>}
</div>
<div className="tb">
  <table class='table' style = {{
    border: "1px solid black",
    textAlign: "center",
    justifyContent: "center",
    width: "50vw"
  }}>
    <thead bgcolor="#000d6c">
      <tr>
        <th style={{border: "1px solid black", color: "white"}}> ITERATION </th>
        <th style={{border: "1px solid black", color: "white"}}> XL </th>
        <th style={{border: "1px solid black", color: "white"}}> XR </th>
        <th style={{border: "1px solid black", color: "white"}}> XM </th>
        <th style={{border: "1px solid black", color: "white"}}> ERROR </th>
      </tr>
    </thead>
    <tbody>
      {!CheckUnValid && Keepvalue.length > 0 && Keepvalue.map((item,index) => {
        return <tr key={index}>
          <td style={{border: "1px solid black"}}>{index}</td>
          <td style={{border: "1px solid black"}}>{item[0]}</td>
          <td style={{border: "1px solid black"}}>{item[1]}</td>
          <td style={{border: "1px solid black"}}>{item[2]}</td>
          <td style={{border: "1px solid black"}}>{item[3]}</td>
        </tr>
      )}}
    </tbody>
  </table>
</div>
</div>
)

```

ตรง className="Showans" คือส่วนที่ไว้แสดงคำตอบที่ได้

หลังจากนั้นตั้งแต่ className="tb" คือการแสดงตาราง โดยแสดงค่าที่ได้ทั้งหมดจากการคำนวณแต่ละรอบ

บทที่ 3

สรุป

3.1 ปัญหาที่พบระหว่างการพัฒนา

1. ไม่มีความรู้ด้าน React จึงต้องมาศึกษาหาความรู้ด้วยตัวเองรวมถึง library ต่างๆ
2. การใช้ map ซึ่งสามารถ ใช้ประโยชน์ได้หลายอย่างแต่ก็ใช้งานได้ยากในการแสดงเว็บต้องมีความเข้าใจในระดับนึงถึงจะเริ่มใช้งานได้
4. ในการทำMatrix นั้นยากมากและการทำงานที่เฉพาะ บังคับใช้ math.js ในการแสดง ค่า และเปลี่ยนค่าต่างๆ

3.2 จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

สามารถคำนวณ Iteration ต่างๆ และแสดงค่ากราฟของแต่ละจุดได้โดยการกำหนดด้วยตัวแปรเดียว

3.3 คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย หรือที่เรียนแล้วไม่เข้าใจ หรืออยากให้เพิ่มสำหรับน้อง ๆ รุ่นต่อไป

อยากให้อาจารย์ช่วยสอน react hook ให้กับรุ่นน้องเพราะ ว่าเป็นการเขียนที่สั้น ประหยัดเวลามากกว่า react แบบเก่าที่เขียนเยอะเป็นพิเศษ และสอนเกี่ยวกับ API ให้มากขึ้น