

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Практична робота № 3
з курсу: «Основи Веб-програмування»

Виконав:
студент 2-го курсу,
групи ТВ-31
Дєдов Данило Романович
<https://github.com/diebymyhand/web-practice>

Перевірив:
Недашківський О.Л.

Київ 2025

Теоретичний матеріал

Для того, щоб порахувати кількість прибутку вдосконалення системи прогнозу, нам спершу треба знайти частку енергії. Вона обчислюється за формулою:

$$\delta_{W1} = \int_x^y p_d dp$$

де:

p_d – значення нормального розподілу потужності;

$W1$ – частка без небалансів;

x, y – діапазон потужності.

Значення нормального розподілу потужності обчислюється за наступною формулою:

$$p_d = \frac{1}{\sigma_1 \sqrt{2\pi}} \exp\left(-\frac{(p - P_c)^2}{2\sigma_1^2}\right)$$

Для перерахунку часток небалансів використовується формула:

$$W = P_c * 24 * \delta_W$$

Прибуток сонячної електростанції обчислюється за формулою:

$$\Pi = W * B$$

де:

B – вартість електроенергії.

Окрім прибутку, варто обрахувати штраф, який виплачує електростанція. Розраховується він за формулою:

$$\Pi = W * B$$

Практична робота № 3

Завдання 1. Створіть веб калькулятор розрахунку прибутку від сонячних електростанцій з встановленою системою прогнозування сонячної потужності.

Опис програмної реалізації

У HTML документі було створено 2 div контейнери. Перший відповідає за відображення полів введення даних, другий – за виведення результатів. При отриманні результатів розрахунків перший контейнер скривається, а другий з'являється на його місці (рис. 1).

```
<main>
  <div id="result" class="container hidden"></div>

  <div id="inputs" class="container">
    <div class="title">Калькулятор</div>

    <label for="power">Потужність:</label>
    <input type="number" id="power" class="input">

    <label for="sigma1">Похибка 1:</label>
    <input type="number" id="sigma1" class="input">

    <label for="sigma2">Похибка 2:</label>
    <input type="number" id="sigma2" class="input">

    <label for="cost">Вартість енергії (грн/кВт*год):</label>
    <input type="number" id="cost" class="input">

    <button type="button" class="button" onclick="calculate()">Розрахувати</button>
  </div>
</main>
```

Рисунок 1 – основний код розмітки сторінки

Завдяки CSS стилям полям калькулятору було надано приємного вигляду. Частину CSS коду зображено на рисунку 2.

```

.container {
  border: 3px solid;
  border-radius: 16px;
  height: 360px;
  width: 20%;
  margin-left: 10px;
  max-width: 300px;
  font-family: "Lucida Console", cursive;
  font-size: 17px;
  margin-left: 100px;
  margin-top: 35px;
}

.hidden {
  display: none;
}

.title {
  font-family: "Lucida Console", cursive;
  font-size: 20px;
  text-align: center;
  margin-top: 15px;
  margin-bottom: 15px;
  text-decoration: underline blue 2px;
}

.button {
  border: solid 2px;
  font-size: 17px;
  font-family: "Lucida Console", cursive;
  background-color: white;
  border-radius: 7px;
  margin-top: 15px;
  margin-left: 17px;
}

```

Рисунок 2 – частина CSS коду

Загалом калькулятор має наступний вигляд (рис. 3).

Калькулятор

Потужність:

Похибка 1:

Похибка 2:

Вартість енергії (грн/
кВт*год):

Рисунок 3 – зображення калькулятора

Коли користувач введе всі вхідні дані та натисне на кнопку «Розрахувати», викликається функція `calculate()`, яка зчитує всі дані з полів вводу та використовує вищенаведені формули для розрахунків (рис. 4).

```
function calculate() {
    let Pc = parseFloat(document.getElementById('power').value);
    let sigma1 = parseFloat(document.getElementById('sigma1').value);
    let sigma2 = parseFloat(document.getElementById('sigma2').value);
    let V = parseFloat(document.getElementById('cost').value);

    if (isNaN(Pc) || isNaN(sigma1) || isNaN(sigma2) || isNaN(V)) {
        alert("Будь ласка, заповніть усі поля!");
        return;
    }

    const delta = 0.05;
    const hoursPerDay = 24;
    const P1 = Pc * (1 - delta);
    const P2 = Pc * (1 + delta);

    const delta_eta1 = integrateNormal(Pc, sigma1, P1, P2);
    const delta_eta2 = integrateNormal(Pc, sigma2, P1, P2);

    const totalEnergy = Pc * hoursPerDay;

    const W1 = totalEnergy * delta_eta1;
    const Pi1 = W1 * V * 1000;
    const W2 = totalEnergy * (1 - delta_eta1);
    const Sh1 = W2 * V * 1000;

    const W3 = totalEnergy * delta_eta2;
    const Pi2 = W3 * V * 1000;
    const W4 = totalEnergy * (1 - delta_eta2);
    const Sh2 = W4 * V * 1000;
    const Pi = Pi2 - Sh2;

    showResult(Pi1, Sh1, Pi2, Sh2, delta_eta1, delta_eta2, Pi);
}
```

Рисунок 4 – код функції `calculate()`

Для обчислення частки енергії було створено функцію `integrateNormal()`, яка інтегрує нормальний розподіл методом прямокутників (рис. 5). Сам нормальний розподіл обраховується у функції `normalPDF()` (рис. 6).

```
function integrateNormal(mu, sigma, lower, upper) {
  let area = 0;
  let h = 0.0001; // крок інтегрування

  for (let i = 0; i < (upper - lower) / h; i++) {
    let x = lower + i * h;
    area += normalPDF(x, mu, sigma) * h; // обчислення площі під графіком
  }

  return area;
}
```

Рисунок 5 – код функції integrateNormal()

```
function normalPDF(x, mu, sigma) {
  return (1 / (sigma * Math.sqrt(2 * Math.PI))) * Math.exp(-0.5 * Math.pow((x - mu) / sigma, 2));
}
```

Рисунок 6 – код функції normalPDF()

У кінці функції для виводу результатів викликається метод showResult(), який ховає контейнер для вводу даних та відображає контейнер з результатами, додаючи до нього HTML код. У доданому коді створюється кнопка «Скинути», при натисканні якої спрацьовує функція reset(). Вона робить зворотні дії: ховає контейнер з результатами та відображає контейнер для вводу даних. Код функцій зображено на рисунку 8.

```
function reset() {
  document.getElementById("result").classList.add("hidden");
  document.getElementById("inputs").classList.remove("hidden");
}

function showResult(Pi1, Sh1, Pi2, Sh2, delta_eta1, delta_eta2, Pi) {
  document.getElementById("inputs").classList.add("hidden");
  let resultContainer = document.getElementById("result");
  resultContainer.classList.remove("hidden");

  resultContainer.innerHTML = `
    <div class="title">Результати</div>
    <div class="result-text"><strong>До покращення:</strong><br></div>
    <div class="result-text">Частка без небалансів: ${((delta_eta1 * 100).toFixed(1))}%<br></div>
    <div class="result-text">Прибуток: ${Pi1.toFixed(1)} грн<br></div>
    <div class="result-text">Штраф: ${Sh1.toFixed(1)} грн</div>

    <div class="result-text"><strong>Після покращення:</strong><br></div>
    <div class="result-text">Частка без небалансів: ${((delta_eta2 * 100).toFixed(1))}%<br></div>
    <div class="result-text">Прибуток: ${Pi2.toFixed(1)} грн<br></div>
    <div class="result-text">Штраф: ${Sh2.toFixed(1)} грн</div>

    <div class="result-text"><strong>Прибуток після покращення (чистими):</strong> ${Pi.toFixed(1)}<br></div>
    <button class="button" onclick="reset()">Скинути</button>
  `;
}
```

Рисунок 8 – код функцій reset() та showResult()

Результати перевірки на контрольному прикладі

Калькулятор

Потужність:

Похибка 1:

Похибка 2:

Вартість енергії (грн/
кВт*год):

Розрахувати

Результати

До покращення:

Частка без небалансів: 19.7%

Прибуток: 165826.6 грн

Штраф: 674173.4 грн

Після покращення:

Частка без небалансів: 68.3%

Прибуток: 573459.2 грн

Штраф: 266540.8 грн

**Прибуток після покращення
(чистими): 306918.3**

Скинути

Висновок

У процесі написання практичної роботи було написано веб-калькулятор для розрахунку срозрахунку прибутку від сонячних електростанцій з встановленою системою прогнозування сонячної потужності Було покращено навички роботи з HTML та CSS. Також покращено роботу з мовою програмування Javascript. Код працює коректно та виконує свої функції.