**Прізвище:** Дацишин

**Ім’я:** Роман

**Група:** КН-405

**Варіант:** 4

**Кафедра:** Кафедра Систем

Автоматизованого Проектування

**Дисципліна:** Теорія прийняття рішень

**Перевірила:** Кривий Р.З.

**Звіт**

До лабораторної роботи №4

На тему “Метод експертної оцінки”

**Мета роботи:** Отримати практичні навички експертного оцінювання систем, різними методами і обробки результатів оцінювання.

**Індивідуальне завдання:**

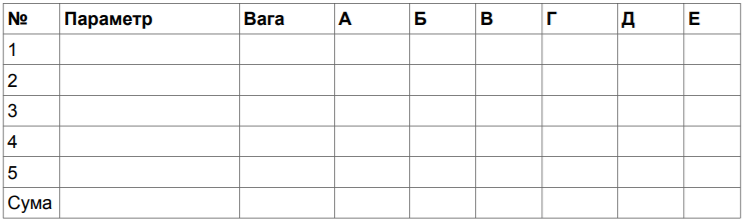
1. Вибираємо об'єкт для експертної оцінки;

2. Вибираємо параметри для порівняння;

3. Визначаємо вагу кожного параметра;

4. Задаємо порівняльну шкалу;

5. Порівнюємо.



**Ручні обрахунки очікуваних доходів**

8\*0.3=2.4 6\*0.3=1.8 4\*0.3=1.2 4\*0.3=1.2 8\*0.3=2.4 2.4

3\*0.2=0.6 3\*0.2=0.6 5\*0.2=1 5\*0.2=1 4\*0.2=0.8 1

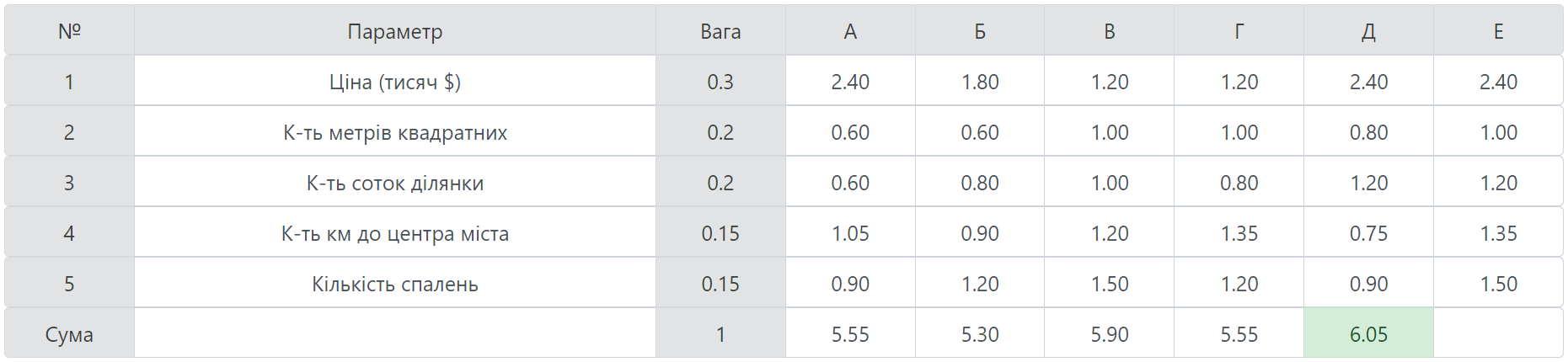
3\*0.2=0.6 4\*0.2=0.8 5\*0.2=1 4\*0.2=0.8 6\*0.2=1.2 1.2

7\*0.15=1.05 6\*0.15=0.9 8\*0.15=1.2 9\*0.15=1.35 5\*0.15=0.75 1.35

6\*0.15=0.9 8\*0.15=1.2 10\*0.15=1.5 8\*0.15=1.2 6\*0.15=0.9 1.5

5.55 5.3 5.9 5.55 **6.05**

**Результат виконання програми**

****

Отже, результати виконання програми збігаються з вище отриманими ручними обрахунками, тому самим оптимальним варіантом буде покупка будинку - Д.

**Код програми:**

**let weights = [];**

**let data = [];**

**let maxValues = [];**

**let sumValues = [];**

**let maxOfSumValues = 0;**

**const showResults = () => {**

**parseData();**

**convertToDecimalMarkSystem();**

**findMaxValues();**

**sumObjectValues();**

**findMaxOfSumValues();**

**pasteTableToTHML();**

**};**

**const parseData = () => {**

**for (let i = 1; i < 6; i++) {**

**weights.push(parseFloat(document.getElementById(`weight\_${i}`).value));**

**data.push({**

**price: parseFloat(document.getElementById(`input${i}\_1`).value),**

**squareMeters: parseFloat(document.getElementById(`input${i}\_2`).value),**

**acresOfLand: parseFloat(document.getElementById(`input${i}\_3`).value),**

**kmToTheСenter: parseFloat(document.getElementById(`input${i}\_4`).value),**

**bedrooms: parseFloat(document.getElementById(`input${i}\_5`).value),**

**});**

**}**

**};**

**const convertToDecimalMarkSystem = () => {**

**for (let i = 0; i < data.length; i++) {**

**data[i].price = (convertPrice(data[i].price) \* weights[0]).toFixed(2);**

**data[i].squareMeters = (**

**convertMeters(data[i].squareMeters) \* weights[1]**

**).toFixed(2);**

**data[i].acresOfLand = (**

**convertAcres(data[i].acresOfLand) \* weights[2]**

**).toFixed(2);**

**data[i].kmToTheСenter = (**

**convertKm(data[i].kmToTheСenter) \* weights[3]**

**).toFixed(2);**

**data[i].bedrooms = (convertbedrooms(data[i].bedrooms) \* weights[4]).toFixed(**

**2**

**);**

**}**

**};**

**const convertPrice = (price) => {**

**if (price < 100) return 10;**

**if (price >= 100 && price < 150) return 8;**

**if (price >= 150 && price < 200) return 6;**

**if (price >= 200 && price < 250) return 4;**

**if (price >= 300 && price < 350) return 2;**

**if (price >= 400) return 1;**

**};**

**const convertMeters = (meters) => {**

**if (meters < 100) return 1;**

**if (meters >= 100 && meters < 120) return 2;**

**if (meters >= 120 && meters < 150) return 3;**

**if (meters >= 150 && meters < 170) return 4;**

**if (meters >= 170 && meters < 200) return 5;**

**if (meters >= 200 && meters < 220) return 6;**

**if (meters >= 220 && meters < 250) return 7;**

**if (meters >= 250 && meters < 270) return 8;**

**if (meters >= 300 && meters < 350) return 9;**

**if (meters >= 350) return 10;**

**};**

**const convertAcres = (acres) => {**

**if (acres < 3) return 2;**

**if (acres >= 3 && acres < 5) return 3;**

**if (acres >= 5 && acres < 8) return 4;**

**if (acres >= 8 && acres < 10) return 5;**

**if (acres >= 10 && acres < 12) return 6;**

**if (acres >= 12 && acres < 15) return 7;**

**if (acres >= 15 && acres < 17) return 8;**

**if (acres >= 17 && acres < 20) return 9;**

**if (acres >= 20) return 10;**

**};**

**const convertKm = (km) => {**

**if (km < 8) return 10;**

**if (km >= 8 && km < 10) return 9;**

**if (km >= 10 && km < 12) return 8;**

**if (km >= 12 && km < 15) return 7;**

**if (km >= 15 && km < 17) return 6;**

**if (km >= 17 && km < 20) return 5;**

**if (km >= 20 && km < 25) return 4;**

**if (km >= 25 && km < 30) return 3;**

**if (km >= 30) return 2;**

**};**

**const convertbedrooms = (bedrooms) => {**

**if (bedrooms == 1) return 2;**

**if (bedrooms == 2) return 4;**

**if (bedrooms == 3) return 6;**

**if (bedrooms == 4) return 8;**

**if (bedrooms > 4) return 10;**

**};**

**const findMaxValues = () => {**

**let maxPrice = -1;**

**let maxMeters = -1;**

**let maxAcres = -1;**

**let maxkm = -1;**

**let maxBedrooms = -1;**

**for (let i = 0; i < data.length; i++) {**

**if (data[i].price > maxPrice) maxPrice = data[i].price;**

**if (data[i].squareMeters > maxMeters) maxMeters = data[i].squareMeters;**

**if (data[i].acresOfLand > maxAcres) maxAcres = data[i].acresOfLand;**

**if (data[i].kmToTheСenter > maxkm) maxkm = data[i].kmToTheСenter;**

**if (data[i].bedrooms > maxBedrooms) maxBedrooms = data[i].bedrooms;**

**}**

**maxValues.push(maxPrice);**

**maxValues.push(maxMeters);**

**maxValues.push(maxAcres);**

**maxValues.push(maxkm);**

**maxValues.push(maxBedrooms);**

**};**

**const sumObjectValues = () => {**

**for (let i = 0; i < data.length; i++) {**

**let sum = 0;**

**sum +=**

**Number(data[i].price) +**

**Number(data[i].squareMeters) +**

**Number(data[i].acresOfLand) +**

**Number(data[i].kmToTheСenter) +**

**Number(data[i].bedrooms);**

**sumValues.push(sum.toFixed(2));**

**}**

**};**

**const findMaxOfSumValues = () => {**

**let max = 0;**

**for (let i = 0; i < sumValues.length; i++) {**

**if (sumValues[i] > max) max = sumValues[i];**

**}**

**maxOfSumValues = max;**

**};**

Повна версія коду доступна на GitHub: [https://github.com/RomanDatsyshyn/TPR](https://github.com/RomanDatsyshyn/TPR/)

**Висновок:** під час виконання цієї лабораторної роботи я одержав практичні навички використання методів експертної оцінки для прийняття рішення. Також було проведено ручні обрахунки і написано програму для вибору будинку цим методом.