**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 №1

На тему “ Прийняття рішень в умовах невизначеності і ризику (на прикладі рішення про дії підприємства для комерційно вигідної стратегії розвитку)”

**Мета роботи:** Одержання практичних навичок використання методів прийняття рішень в умовах невизначеності і ризику.

**Індивідуальне завдання:**

Задача. Опис

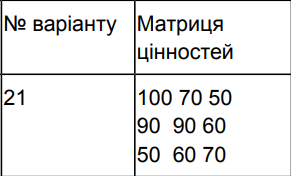
Компанія має три альтернативних варіанти своєї стратегії розвитку. Оцінка його прибутку в залежності від стану зовнішнього середовища наведено в таблиці .

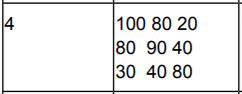
**А) Прийняти рішення в умовах невизначеності.**

Необхідно знайти оптимальні стратегії при песимістичній оцінці (по критерію Вальда), оцінці Лапласа, по критерію Гурвіца. Значення коефіцієнта оптимізму вибрати самостійно. Результати вибору рішення відобразити в таблиці. Зробити висновки по застосуванню критеріїв

**Б) Прийняти рішення в умовах ризику**

Нехай отримані експертні оцінки ймовірностей стану зовнішнього середовища p1=0.5, p2=0.35, p3=0.15. Оцінити альтернативні рішення по критерію БайесаЛапласа. Результати обчислень цінностей альтернативних рішень занести в туж таблицю. Вибрати найкраще рішення. Порівняти результати вибору з отриманими раніше результатами вибору рішення в умовах невизначеності.

****

****

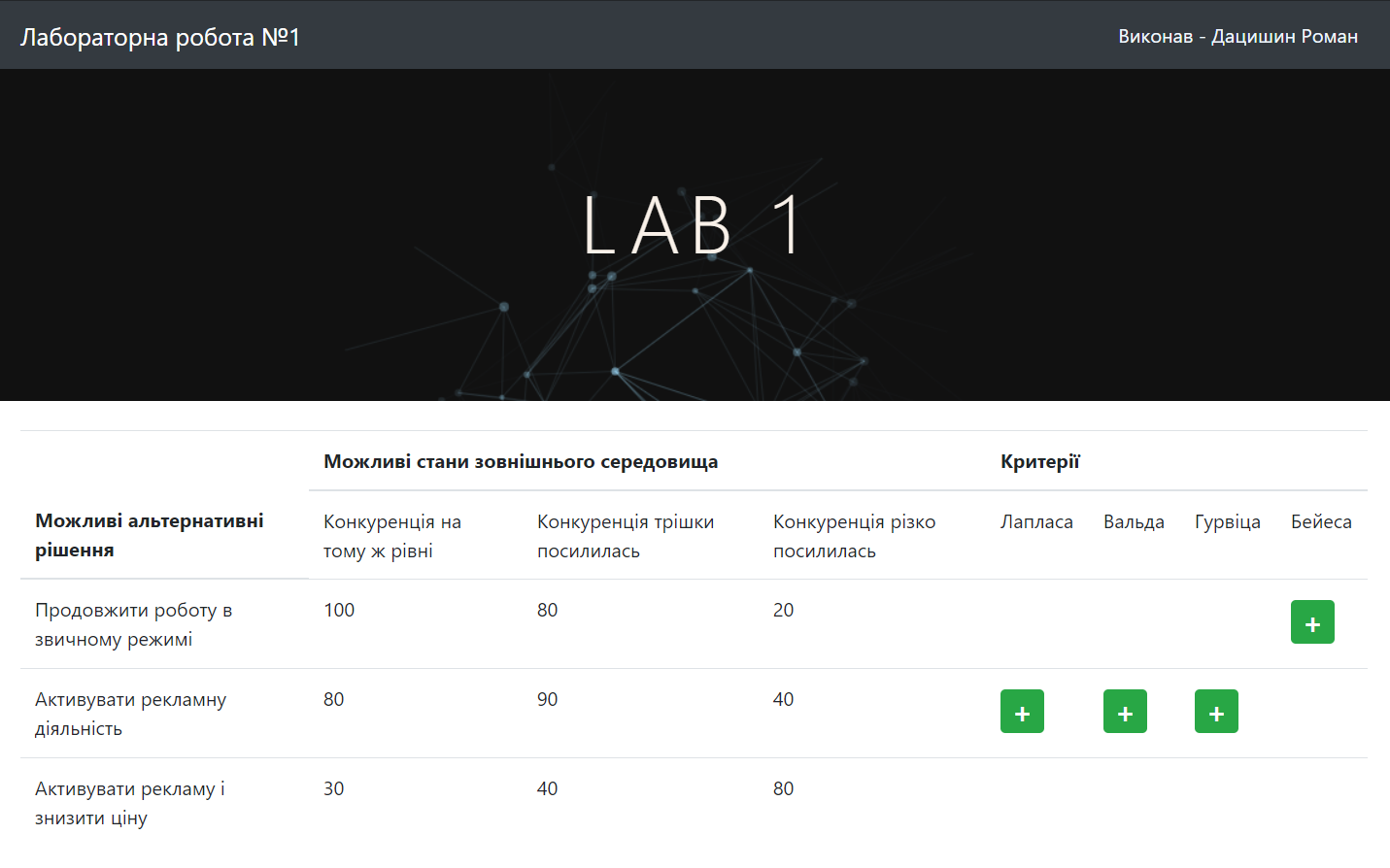
****

Рис. 1 Результат виконання програми

Отже, враховуючи отримані результати програми можна сказати, що найвигіднішою стратегією розвитку є активувати рекламну діяльність, оскільки ця стратегіє виграє по трьох критеріях. Стратегія – продовжити роботу в звичному режимі виграє по одному критерію, критерію Байеса.

Для критерія Гурвіца було використано коефіцієнт оптимізму 0.5

**Код програми:**

**const WaldCriteria = (matrix) => {**

**let lowestValues = [];**

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

**let compareWith = 999;**

**for (let j = 0; j < matrix.length; j++) {**

**if (matrix[i][j] < compareWith) {**

**compareWith = matrix[i][j];**

**}**

**}**

**lowestValues.push(compareWith);**

**}**

**let highestNumber = { Option: 1, Value: lowestValues[0] };**

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

**if (lowestValues[i] > highestNumber.Value) {**

**highestNumber = { Option: i + 1, Value: lowestValues[i] };**

**}**

**}**

**return highestNumber;**

**};**

**const LaplaceCriteria = (matrix) => {**

**let sums = [];**

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

**let sum = 0;**

**for (let j = 0; j < matrix.length; j++) sum += matrix[i][j];**

**sums.push(sum / matrix.length);**

**}**

**let highestNumber = { Option: 1, Value: sums[0] };**

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

**if (sums[i] > highestNumber.Value) {**

**highestNumber = { Option: i + 1, Value: sums[i] };**

**}**

**}**

**return highestNumber;**

**};**

**const HurwitzCriteria = (matrix) => {**

**let lowestValues = [];**

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

**let compareWith = 999;**

**for (let j = 0; j < matrix.length; j++) {**

**if (matrix[i][j] < compareWith) {**

**compareWith = matrix[i][j];**

**}**

**}**

**lowestValues.push(compareWith);**

**}**

**let highestValues = [];**

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

**let compareWith = -999;**

**for (let j = 0; j < matrix.length; j++) {**

**if (matrix[i][j] > compareWith) {**

**compareWith = matrix[i][j];**

**}**

**}**

**highestValues.push(compareWith);**

**}**

**let values = [];**

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

**values.push(0.5 \* lowestValues[i] + (1 - 0.5) \* highestValues[i]);**

**}**

**let highestNumber = { Option: 1, Value: values[0] };**

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

**if (values[i] > highestNumber.Value) {**

**highestNumber = { Option: i + 1, Value: values[i] };**

**}**

**}**

**return highestNumber;**

**};**

**const BayesLaplaceCriteria = (matrix) => {**

**let p = [0.5, 0.35, 0.15];**

**let values = [];**

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

**values.push(**

**matrix[i][0] \* p[0] + matrix[i][1] \* p[1] + matrix[i][2] \* p[2]**

**);**

**}**

**let highestNumber = { Option: 1, Value: values[0] };**

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

**if (values[i] > highestNumber.Value) {**

**highestNumber = { Option: i + 1, Value: values[i] };**

**}**

**}**

**return highestNumber;**

**};**

GitHub: <https://github.com/RomanDatsyshyn/TPR/>

**Висновок:** під час виконання цієї лабораторної роботи я одержав практичні навички використання методів прийняття рішень в умовах невизначеності і ризику. Також було написано програму для обчислення критеріїв Вальда, Лапласа, Гурвіца та Бейеса. За результатами виконання програми було визначено стратегію розвитку – активувати рекламну діяльність.