ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ

з дисципліни «Основи програмування на мові Javascript»

студента групи КС-21-2

Губа Максім

кафедра комп’ютерних технологій, ДНУ 2024/2025

**Постановка задачі:** Здійснити перестановку значень змінних A,B,C,D,E в змінні AEBDC відповідно.

**Опис розв’язку:**

Для виконання перестановки значень змінних A, B, C, D, E у відповідний порядок AEBDC, спочатку програма приймає введення значень для кожної змінної. Щоб гарантувати коректне введення, реалізована перевірка на порожні значення. Якщо користувач не введе значення для змінної, програма повторно запитує введення.

Після того, як всі значення змінних отримано, їх зберігають у початковому стані для подальшого відображення. Використовуючи тимчасову змінну temp, здійснюється обмін значеннями між змінними згідно з правилами перестановки. Наприклад, значення змінної E перепризначається до змінної B, а значення D змінюється на значення C.

У кінці програма виводить повідомлення, в якому показані початкові та кінцеві значення змінних після перестановки.

**Вихідний текст програми розв’язку задачі (текст)**

// Функція для перестановки значень змінних  
function rearrangeVariables(order) {  
 // Введення значень для змінних  
 let A = prompt("Введіть значення для A:");  
 let B = prompt("Введіть значення для B:");  
 let C = prompt("Введіть значення для C:");  
 let D = prompt("Введіть значення для D:");  
 let E = prompt("Введіть значення для E:");  
  
 // Зберігаємо значення змінних у об'єкті  
 let variables = { A, B, C, D, E };  
  
 // Створюємо новий масив відповідно до заданого порядку  
 let reordered = order.split("").map(key => variables[key]);  
  
 // Виводимо результат  
 ***console***.log("Новий порядок змінних:", reordered);  
 alert("Новий порядок змінних: " + reordered.join(", "));  
}  
  
// Приклад виклику функції з порядком "AEBDC"  
rearrangeVariables("AEBDC");

**Опис інтерфейсу програми (керівництво користувача):**

Після запуску програми користувачеві послідовно з'являються вікна із запитом на введення значень для змінних A, B, C, D, E.

У кожному вікні необхідно ввести відповідне значення для змінної. Після успішного введення всіх значень, програма автоматично виконає перестановку.

Результат перестановки відображається у вигляді спливаючого повідомлення, яке містить змінені значення змінних.

**Опис тестових прикладів:**

Тест 1:

Вхід:

A = 1, B = 2, C = 3, D = 4, E = 5

Очікуваний результат:

A: 1, B: 5, C: 2, D: 4, E: 3

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

**Тест 2:**

Вхід:

A = "a", B = "b", C = "c", D = "d", E = "e"

Очікуваний результат: AEDCB

A: "a", B: "e", C: "b", D: "d", E: "c"

Изображение выглядит как текст, снимок экрана, Шрифт, логотип

Автоматически созданное описание Усі тестові приклади проходять без помилок та забезпечують правильний результат.

**Аналіз помилок (опис усунення зауважень):**

Під час тестування програми виявлено кілька проблем, пов'язаних із введенням порожніх значень. Спочатку програма не перевіряла на наявність порожніх рядків, через що виникала помилка при передачі значень змінних. Для усунення цієї проблеми було додано валідацію введення. Якщо користувач не вводить значення, програма повідомляє про помилку і повторює запит до моменту введення непорожнього значення.

**Завдання 2**

**Постановка задачі**: Для геометричної фігури, заданої на дійсній площині, обчислити певний параметр фігури, або відобразити фігуру симетрично відносно вказаної точки або вказаної прямої лінії (вертикальні)

**Опис розв’язку:**

Задача полягає в тому, щоб відобразити трикутник відносно осі ординат (ось х). Для цього було реалізовано функцію, яка приймає координати трьох вершин трикутника і відображає їх. Відображення по осі y передбачає зміну знака координати x для кожної точки, при цьому координата y залишається незмінною.

Спочатку користувач вводить координати вершин трикутника через послідовні запити у вікні. Дані вводяться для кожної точки (A, B, C). Щоб уникнути некоректного введення, додана валідація значень на випадок, якщо користувач введе нечислові дані. Якщо введення не відповідає числовим значенням, програма повторює запит.

Після отримання початкових координат трикутника, вони передаються у функцію відображення, яка змінює знак x для кожної точки. Після виконання операції початкові та відображені координати виводяться на екран у вигляді повідомлення.

**Вихідний текст програми розв’язку задачі :**// Завдання 2: Відображення трикутника відносно осі абсцис  
  
// Введення координат вершин трикутника  
let ***x1*** = parseFloat(prompt("Введіть x1:"));  
let ***y1*** = parseFloat(prompt("Введіть y1:"));  
let ***x2*** = parseFloat(prompt("Введіть x2:"));  
let ***y2*** = parseFloat(prompt("Введіть y2:"));  
let ***x3*** = parseFloat(prompt("Введіть x3:"));  
let ***y3*** = parseFloat(prompt("Введіть y3:"));  
  
// Вивід початкових координат  
***console***.log("Початкові координати трикутника:");  
***console***.log(`Точка A: (${***x1***}, ${***y1***})`);  
***console***.log(`Точка B: (${***x2***}, ${***y2***})`);  
***console***.log(`Точка C: (${***x3***}, ${***y3***})`);  
  
// Відображення відносно осі абсцис  
***y1*** = -***y1***;  
***y2*** = -***y2***;  
***y3*** = -***y3***;  
  
// Вивід нових координат  
***console***.log("Координати трикутника після відображення:");  
***console***.log(`Точка A: (${***x1***}, ${***y1***})`);  
***console***.log(`Точка B: (${***x2***}, ${***y2***})`);  
***console***.log(`Точка C: (${***x3***}, ${***y3***})`);  
alert(`Координати після відображення:\nA(${***x1***}, ${***y1***})\nB(${***x2***}, ${***y2***})\nC(${***x3***}, ${***y3***})`);

**Опис інтерфейсу програми (керівництво користувача):**

1. Після запуску програми користувачеві з'являються запити на введення координат для трьох точок трикутника: A, B, C.
2. Після введення всіх координат, програма автоматично відобразить трикутник по осі х.
3. Результати відображення трикутника виводяться у вигляді спливаючого повідомлення, яке містить початкові та відображені координати точок трикутника.

**Опис тестових прикладів:**

**Тест 1:**  
Вхід:  
A(1, 2), B(3, 4), C(5, 6)  
Очікуваний результат:  
Початкові координати: A(1, 2), B(3, 4), C(5, 6)  
Відображені координати: A(1, -2), B(3, -4), C(5, -6)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Тест 2:**  
Вхід:  
A(1, 1), B(3, -4), C(5, -6)  
Очікуваний результат:  
Початкові координати: A(0, 0), B(2, 3), C(4, -5)  
Відображені координати: A(0, 0), B(-2, 3), C(-4, -5)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Усі тестові приклади успішно виконуються без помилок.

**Аналіз помилок (опис усунення зауважень):**

Під час тестування програми було виявлено проблему з введенням некоректних значень. Наприклад, якщо користувач залишав поле для введення порожнім або вводив текст, програма видавала помилку і завершувала виконання.

**Завдання 3**

**Постановка задачі:** Задана пара геометричних фігур на дійсній площині і задано точку. Треба визначити, чи потрапляє точка у перетин (чи в об’єднання) вказаних фігур

**Опис розв’язку:**

Задача полягає в тому, щоб перевірити, чи потрапляє точка у задані вертикальну або вертикальна смужки. Для цього програма отримує від користувача параметри вертикальної та горизонтальної смужки (їхні межі) і координати точки, а потім перевіряє, чи знаходиться ця точка у одній з цих смужок.

**Вихідний текст програми розв’язку задачі :**// Завдання 3: Чи належить точка об'єднанню двох вертикальних стрічок  
  
// Введення координат стрічок та точки  
let ***x1\_1*** = parseFloat(prompt("Введіть x1 для першої стрічки:"));  
let ***x2\_1*** = parseFloat(prompt("Введіть x2 для першої стрічки:"));  
  
let ***x1\_2*** = parseFloat(prompt("Введіть x1 для другої стрічки:"));  
let ***x2\_2*** = parseFloat(prompt("Введіть x2 для другої стрічки:"));  
  
let ***x*** = parseFloat(prompt("Введіть координату x точки:"));  
let ***y*** = parseFloat(prompt("Введіть координату y точки:"));  
  
// Перевірка належності точки об'єднанню стрічок  
const ***inFirstStripe*** = ***x1\_1*** <= ***x*** && ***x*** <= ***x2\_1***; // Належить першій стрічці  
const ***inSecondStripe*** = ***x1\_2*** <= ***x*** && ***x*** <= ***x2\_2***; // Належить другій стрічці  
  
// Присвоюємо значення змінній result  
const ***result*** = ***inFirstStripe*** || ***inSecondStripe***; // Належить хоча б одній із стрічок?  
  
// Вивід результату  
***console***.log("Початкові дані:");  
***console***.log(`Перша стрічка: x1=${***x1\_1***}, x2=${***x2\_1***}`);  
***console***.log(`Друга стрічка: x1=${***x1\_2***}, x2=${***x2\_2***}`);  
***console***.log(`Точка: (${***x***}, ${***y***})`);  
  
if (***result***) {  
 ***console***.log("Точка належить об'єднанню стрічок.");  
 alert("Точка належить об'єднанню стрічок.");  
} else {  
 ***console***.log("Точка НЕ належить об'єднанню стрічок.");  
 alert("Точка НЕ належить об'єднанню стрічок.");  
}

**Опис інтерфейсу програми (керівництво користувача):**

1. **Запуск програми**

* Запустіть програму у браузері (через консоль інструментів розробника або як HTML-файл).

1. **Введення даних**

* Користувач вводить **межі двох вертикальних стрічок** (інтервали xxx-координат):
  + **x1** (ліва межа першої стрічки).
  + **x2** (права межа першої стрічки).
  + **x3** (ліва межа другої стрічки).
  + **x4** (права межа другої стрічки).
* Після цього користувач вводить координати **точки**:
  + xxx-координату.
  + yyy-координату.

1. **Валідація вводу**

* Програма перевіряє, чи всі введені значення є числовими.
* Якщо введені дані некоректні, програма просить повторити введення.

1. **Обчислення результату**

* Програма перевіряє, чи xxx-координата точки належить хоча б одній із двох стрічок.

1. **Виведення результату**

* Якщо точка належить об'єднанню стрічок:  
  **"Точка належить об'єднанню стрічок."**
* Якщо точка **не належить** об'єднанню стрічок:  
  **"Точка НЕ належить об'єднанню стрічок."**
* Результат виводиться у вікні повідомлення (alert) та у консолі браузера.

**Опис тестових прикладів:**

**Тест 1:**  
Вхідні дані:

* Перша стрічка: x1=1, x2=3
* Друга стрічка: x1=5, x2=7
* Точка: x=4, y =2 **Изображение выглядит как текст, снимок экрана, Шрифт, логотип

  Автоматически созданное описание**

**Тест 2:**  
Вхідні дані:

Перша стрічка: x1=1, x2=1

Друга стрічка: x1=2, x2=2

Точка: (2, 3) Изображение выглядит как текст, снимок экрана, Шрифт, логотип

Автоматически созданное описание

**Аналіз помилок (опис усунення зауважень):**

Основні проблеми під час тестування виникали через введення користувачем некоректних значень, таких як текст замість чисел, або залишення полів порожніми. Щоб уникнути цих ситуацій, у програму було додано валідацію введених даних. Тепер кожне значення перевіряється на відповідність числовому типу, і якщо введене значення некоректне, програма запитує його повторно.

Під час тестування також перевірялися випадки з від’ємними координатами та точками, що знаходяться на межах смужок. Програма успішно обробляє ці випадки, коректно розпізнаючи точки, що належать смужкам, або такі, що знаходяться поза їх межами.

Додатково були протестовані сценарії з нульовими значеннями координат. Програма працює стабільно і правильно обробляє точки зі значенням 0.

Усі виявлені під час тестування проблеми були усунені, і програма успішно проходить перевірку для всіх сценаріїв введення.