

**Course Experiment Report**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course:** | Java Language | | | | | | |
|  |  | | | | | | |
| **Semester:** | 1-18th | **week** | 2nd | **year** | | 1st | **term** |
|  |  |  |  |  | |  |  |
| **Major:** | Software Engineering | | | | | **Class:** | 2020 |
|  |  | | | | |  |  |
| **Student name:** | 温长锟 | | **Student No.:** | | 222020321062106 | | |
|  |  | |  | |  | | |
| **Teacher:** | 詹成 | | | | | | |

College of Computer and Information Science

|  |  |  |  |
| --- | --- | --- | --- |
| Project | Exp3 Array | | |
| Time | 2021.10.14 | Type | □Verification □Design □Synthetical |
| 1. Answer the questions   (1) What is the difference between array variables and primitive data type variables?  Answer：The value of array variable is a memory address, corresponding to an area in heap memory; The original variables are stored in stack memory  (2) How to determine the location of the 3 \* 3 box in above experiment.  Answer：Divide the Sudoku grid into nine large parts  judge by dividing the number of subscript serial numbers by three（[i/3][j/3]）  (3) How to understand that a two-dimensional array is an array of one-dimensional arrays.  Answer:  There is no difference in their storage  Schematic diagram of two-dimensional array in memory  IMG_256  (4) Other experience.  This experiment has only three short classes. Although the time is very short, I gain a lot every minute.  It greatly deepened my understanding of some theoretical knowledge. I have a deeper understanding of two-dimensional arrays.  It not only gave me a new understanding of Java in theory, but also improved my practical ability, and really put what I learned into practice.  2. All Codes  import java.util.Scanner;  public class Sudo {   public static void main(String[] args) {  Scanner input = new Scanner(System.*in*);  int[][] grid = {{5,3,0,0,7,0,0,0,0},  {6,0,0,1,9,5,0,0,0},  {0,9,8,0,0,0,0,6,0},  {8,0,0,0,6,0,0,0,3},  {4,0,0,8,0,3,0,0,1},  {7,0,0,0,2,0,0,0,6},  {0,6,0,0,0,0,0,0,0},  {0,0,0,4,1,9,0,0,5},  {0,0,0,0,8,0,0,7,9}};  *printGrid*(grid);  while(*countRemain*(grid) != 0)  {  System.*out*.print("Enter row, column and number([1-9] [1-9] [1-9]):");  int row = input.nextInt();  int col = input.nextInt();  int num = input.nextInt();  if (*isValid*(grid, row - 1, col - 1, num))  grid[row - 1][col - 1] = num;  *printGrid*(grid);  }  System.*out*.println("You win!");  input.close();  }   public static void printGrid(int grid[][])  {  for (int i = 0; i < 9; i++)  {  for (int j = 0; j < 9; j++)  {  System.*out*.printf(" %d", grid[i][j]);  if (j == 2 || j ==5)  System.*out*.print(" |");  }  System.*out*.println();  if (i == 2 || i == 5)  System.*out*.println("-------|-------|-------");  }  }   public static int countRemain(int[][] grid)  {  int count = 0;  for (int i = 0; i < 9; i++)  for(int j = 0; j < 9; j++)  if (grid[i][j] == 0)  count++;  return count;  }   public static boolean isValid(int grid[][], int row, int col, int num)  {  if (num < 1 || num > 9 || row < 0 || row > 8 || col < 0 || col > 8)  {  System.*out*.println("Invalid number, try again");  return false;  }  if (grid[row][col] != 0)  {  System.*out*.println("not empty, try again");  return false;  }  for (int i = 0; i < 9; i++)  if (grid[i][col] == num)  {  System.*out*.println("same number in this column, try again");  return false;  }  for (int j = 0; j < 9; j++)  if (grid[row][j] == num)  {  System.*out*.println("same number in this row, try again");  return false;  }  for (int i = row - row % 3; i < row - row % 3 + 3; i++)  for (int j = col - col % 3; j < col - col % 3 + 3; j++)  if (grid[i][j] == num)  {  System.*out*.println("same number in this block, try again");  return false;  }  return true;  }  Experimental Result：run  Enter the right number    Enter the false number: | | | |

|  |  |  |
| --- | --- | --- |
| Evaluation | Code Correctness (60%): |  |
| Experience (40%): |  |
| Score： | |