

**Course Experiment Report**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course:** | Java Language | | | | | | |
|  |  | | | | | | |
| **Semester:** | 1-18th | **week** | 2nd | **year** | | 1st | **term** |
|  |  |  |  |  | |  |  |
| **Major:** | Software Engineering | | | | | **Class:** | 2019 |
|  |  | | | | |  |  |
| **Student name:** | Fu Ruoxuan | | **Student No.:** | | 222019321062060 | | |
|  |  | |  | |  | | |
| **Teacher:** | Wang Xiaomeng | | | | | | |

College of Computer and Information Science

|  |  |  |  |
| --- | --- | --- | --- |
| Project | Exp3 Array | | |
| Time | 10/14/2020 | Type | □Verification □Design □Synthetical |
| 1. Answer the questions  (1) What is the difference between array variables and primitive data type variables?  An array is a set of variables which are the same data type. We need to use the for loop to traverse every element to operate it. When we copy an array, we can not simply use “arr2=arr1”.Array name is considered as a reference. Therefore the statement means that we let arr2 refer to the memory space of arr1.To avoid this, We should use loops to copy every elements or a static method called arraycopy(). What is more, while we pass the array name to a method, we are passing the reference. So what we operate on array in this method, it will affect this array.  (2) How to determine the location of the 3 \* 3 box in above experiment.  For any elements, the starting elements of the box is grid[3\*(i/3)][3\*(j/3)] while the element is grid[i][j].  (3) How to understand that a two-dimensional array is an array of one-dimensional arrays.  We can consider the two-dimensional array as a one-dimensional array which has a set of arrays which have the same datatype.  (4) Other experience.  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);  int remain = countRemainNum(grid);  // Play the game  while(remain > 0){  System.out.print("Enter row, column and number([1-9] [1-9] [1-9]):");  int r = input.nextInt() - 1;  int c = input.nextInt() - 1;  int n = input.nextInt();  if(grid[r][c] != 0){  System.out.println("The cell is not empty!");  }else{  if(isValid(r,c,n,grid)){  grid[r][c] = n;  remain--;  System.out.println("The current state of grid:");  printGrid(grid);  }else{  System.out.println("The number is irrational.");  }  }  }  // End the game and print the final solution  // Close the input stream  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+1)%3==0){  System.out.print("| ");  }  }  System.out.println();  if ((i+1)%3==0){  System.out.print("----- | ----- | ----- |\n");  }  }  }  public static int countRemainNum(int[][] grid) {  int remain = 0;  for (int i=0;i<9;i++){  for(int j = 0;j<9;j++){  if(grid[i][j]==0){  remain++;  }  }  }  return remain;  }  public static boolean isValid(int r, int c, int n, int[][] grid) {  // Check whether n is valid at the r-1 row  for (int column = 0; column < 9; column++) {  if (grid[r][column] == n) {  return false;  }  }  // Check whether n is valid at the c-1 column  for (int row = 0; row<9;row++ ){  if (grid[row][c]==n){  return false;  }  }  // Check whether n is valid in the 3 by 3 box  for (int row=(3\*(r/3));row<(3\*(r/3))+3;row++){  for(int column=(3\*(c/3));column<(3\*(c/3))+3;column++){  if (grid[row][column]==n){  return false;  }  }  }  return true; // The value n at grid[i][j] is valid  }  } | | | |
|  | | | |

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