

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

**Лабораторная работа №1**

По дисциплине: «Современные платформы программирования»

**Выполнил:**  
студент 3 курса  
группы ПО-8  
Сорока В.С.

**Проверил:**  
Крощенко А.А.

Брест, 2024

**Цель работы:** приобрести практические навыки обработки параметров командной строки, закрепить базовые знания языка программирования Java при решении практических задач.

## Задание 1 (Вариант 8)

Вывод всех уникальных чисел

### Спецификации ввода-вывода программы:

- Программа генерирует случайный массив из 5 целых чисел.
- Сгенерированные числа находятся в диапазоне от 1 до 10 (включительно).
- Выводится исходный массив в формате [число1, число2, число3, число4, число5].
- Пример вывода: Исходный массив: [4, 7, 2, 10, 6].
- Программа вызывает функцию printResult и передает сгенерированный массив в качестве аргумента.
- Функция printResult принимает массив целых чисел и обрабатывает его.
- Выводится список уникальных чисел из исходного массива.
- Уникальные числа выводятся через пробел.
- Пример вывода: Уникальные числа: 4 7 2 10 6.
- Дубликаты чисел (числа, встречающиеся более одного раза) не выводятся.
- Функция printResult создает Map<Integer, Integer>, где ключ - это уникальное число, а значение - количество его появлений в массиве.
- Программа обрабатывает каждый элемент массива и увеличивает его значение в Map, если число уже имеется в Map, или добавляет его со значением 1, если число еще не присутствует.
- Затем происходит итерация по Map и выводятся только те числа, у которых значение равно 1, т.е. уникальные числа

### Текст программы

```
public static void printResult(int[] array)
{
    Map<Integer, Integer> uniqueNumbers = new HashMap<Integer, Integer>();
    for (int i = 0; i < array.length; i++)
    {
        if (uniqueNumbers.containsKey(array[i]))
        {
            uniqueNumbers.put(array[i], uniqueNumbers.get(array[i]) + 1);
        } else {
            uniqueNumbers.put(array[i], 1);
        }
    }
    System.out.println("Уникальные числа:");
    for (Map.Entry<Integer, Integer> item : uniqueNumbers.entrySet())
    {
        if (item.getValue() == 1)
        {
            System.out.print(item.getKey() + " ");
        }
    }
}
```

```

case 1:
    Random rand = new Random();
    int[] array = new int[5];
    for (int i = 0; i < 5; i++)
    {
        array[i] = rand.nextInt(10) + 1;
    }
    System.out.println("Исходный массив:\n" + Arrays.toString(array));
    printResult(array);
    break;

```

## Пример работы программы

```

1. Задание 1
2. Задание 2
3. Задание 3
4. Выход

Выберите необходимое: 1

=====
Исходный массив:
[1, 9, 4, 1, 10]
Уникальные числа:
4 9 10
=====

```

## Задание 2 (Вариант 11)

Напишите метод `double[][] single(int size)`, который создает единичную матрицу заданного размера.

### Спецификации ввода-вывода программы:

- Программа запрашивает ввод размера единичной матрицы.
- Ввод должен быть целым числом.
- Размер единичной матрицы представляет собой одно целое число.
- Размер должен быть положительным и не превышать 20.
- При некорректном вводе (отрицательные числа, нулевой размер, размер больше 20) программа сообщает об ошибке и просит повторить ввод.
- Программа вызывает функцию `createIdentityMatrix` и передает введенный размер в качестве аргумента.
- Функция `createIdentityMatrix` создает и возвращает единичную матрицу заданного размера.
- Выводится единичная матрица.
- Элементы на главной диагонали матрицы равны 1, а все остальные элементы равны 0.

- Матрица выводится в формате таблицы с пробелами или разделителями между элементами.
- Пример вывода при размере 3:

```
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
```

## Текст программы

```
public static double[][] createIdentityMatrix(int size)
{
    double[][] matrix = new double[size][size];
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            if (i == j)
            {
                matrix[i][j] = 1.0;
            } else {
                matrix[i][j] = 0.0;
            }
        }
    }
    return matrix;
}
```

```
case 2:
    int size;
    while (true)
    {
        System.out.print("Введите размер единичной матрицы: ");
        size = in.nextInt();
        // Проверка на отрицательные числа, 0 и числа не более 20
        if (size <= 0 || size > 20)
        {
            System.out.println("Размер единичной матрицы должен быть
положительным и не превышать 20. Повторите ввод.\n");
            continue;
        }
        break;
    }
    double[][] matrix = createIdentityMatrix(size);
    System.out.println("Единичная матрица:\n");
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            System.out.print(matrix[i][j] + " ");
        }
        System.out.println();
    }
    break;
```

## Пример работы программы:

```
1. Задание 1
2. Задание 2
3. Задание 3
4. Выход

Выберите необходимое: 2

=====
Введите размер единичной матрицы: 3
Единичная матрица:

1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0

=====
```

### Задание 3 (Вариант 11)

Напишите метод `String capitalize(String str)`, выполняющий капитализацию строки. Капитализация – это замена первых букв слов с маленьких на большие.

Метод должен работать следующим образом:

`capitalize ( null ) = null`

`capitalize ( "" ) = ""`

`capitalize ( " cat " ) = " Cat"`

`capitalize ( " cAt " ) = " CAt"`

### Спецификации ввода-вывода программы:

- Программа запрашивает ввод строки с клавиатуры.
- Ввод может содержать любые символы, включая пробелы и специальные символы.
- Ввод должен быть в виде строки.
- Выполняется вызов функции `capitalize` с введенной строкой в качестве аргумента.
- Функция `capitalize` преобразует строку, капитализируя первую букву каждого слова.
- Если вводная строка равна `null` (без учета регистра), функция возвращает значение `null`.
- Если вводная строка является пустой строкой, функция возвращает пустую строку.

- Если вводная строка содержит только значение null (игнорируя регистр), функция также возвращает значение null.
- Результирующая капитализированная строка выводится на экран.
- Пример вывода:  
Введите строку: hello world  
Капитализированная строка: Hello World

### Текст программы

```
public static String capitalize(String str)
{
    if (str == null) {
        return null;
    }
    if (str.equalsIgnoreCase("null")) {
        return str;
    }
    if (str.isEmpty()) {
        return "";
    }
    String[] words = str.trim().split(" ");
    StringBuilder capitalizedStr = new StringBuilder();
    for (String word : words) {
        if (word != null && !word.isEmpty()) {
            char firstChar = Character.toUpperCase(word.charAt(0));
            String restOfWord = word.substring(1);
            String capitalizedWord = firstChar + restOfWord;
            capitalizedStr.append(capitalizedWord).append(" ");
        }
    }
    return capitalizedStr.toString().trim();
}
```

```
case 3:
    Scanner scanner = new Scanner(System.in);
    System.out.print("Введите строку: ");
    String str = scanner.nextLine();
    String capitalizedStr = capitalize(str.trim());
    System.out.println("Капитализированная строка: " + capitalizedStr);
    break;
```

### Пример работы программы

```
1. Задание 1
2. Задание 2
3. Задание 3
4. Выход

Выберите необходимое: 3

=====
Введите строку: helLo wOrlD
Капитализированная строка: HelLo WOrlD
=====
```

**Вывод:** получил практические навыки обработки параметров командной строки, закрепил базовые знания языка программирования Java при решении практических задач.