

ОРГАНИЗАЦИЯ И ОБРАБОТКА ДАННЫХ В ВИДЕ ТАБЛИЦ В JAVA.

© **Шадрина Наталья Николаевна**

старший преподаватель,

Бурятский государственный университет имени Доржи Банзарова

Россия, 670000, г. Улан-Удэ, ул. Смолина, 24а

E-mail: shadrinann8@yandex.ru

В статье рассмотрены различные варианты организации и обработки данных в виде таблиц. Особое внимание уделено способу организации простых таблиц с использованием конструкторов объекта JTable. Рассмотрен объект JTable, и связанные с ним интерфейсы TableModel, TableColumnModel, SelectionModel, позволяющие строить сложные и объемные таблицы. Рассмотрены также конструкторы объекта JTable, позволяющие строить простые таблицы с возможностью различных вариантов представления данных, с размещением как в панели прокрутки, так и напрямую на форме. Приведены примеры управления компонентами. Подробно рассмотрен пример построения простой таблицы на основе конструкторов объекта JTable.

Ключевые слова. Программирование на Java, объект JTable, конструкторы и интерфейсы объекта JTable.

В зависимости от задачи, которая стоит перед программистом, исполнитель выбирает для себя оптимальные способы представления данных в своем проекте. Возможности выбора достаточно большие – данные можно представить, например, в виде списков, очередей, деревьев, множеств, таблиц и т.д. В данной статье рассмотрен один из способов представления данных, а именно организация данных в виде таблиц. Этот способ отличается достаточной наглядностью и удобством обработки.

В языке программирования Java существует исходный объект JTable, который как раз и позволяет отобразить двухмерную информацию в виде строк и столбцов. Помимо отображения данных в комфортной для восприятия форме, в образованных на основе объекта JTable таблицах есть возможности сортировать и настраивать данные, выводить их в различных видах. Также можно задавать заголовки таблицы и совершать разнообразные действия с ее выделенными элементами.

Для продуктивной работы с таблицами JTable использует три модели и несколько конструкторов. Каждая модель реализует специальные интерфейсы, выполняющие свои особые функции. Эти модели являются источником данных JTable и при изменении данных сохраняют все изменения.

Для хранения данных, содержащихся в ячейках таблицы, используется интерфейс TableModel. Модель, реализующая интерфейс TableModel, хранит также дополнительную служебную информацию об этих ячейках.

Для управления столбцами таблицы используется вторая модель. Она реализует интерфейс TableColumnModel. С помощью интерфейса TableColumnModel выполняются различные манипуляции со столбцами – их можно добавлять, перемещать, менять местами. Можно установить размеры столбцов, изменить расстояние между ними.

Для работы со строками предназначена третья модель таблицы, которая реализует интерфейс `SelectionModel`. Этот интерфейс отвечает за выделение строк и работу с выделенными строками.

Однако, помимо моделей, которые целесообразно подключать для создания сложных и объемных таблиц, `JTable` может использовать собственные конструкторы. В Java существует достаточное разнообразие конструкторов `JTable`, позволяющее создавать простые таблицы с различными видами инициализаций, с определенным количеством колонок и строк и т.д. Для того, чтобы создать простую таблицу, данные необходимо представить в массивах или векторах.

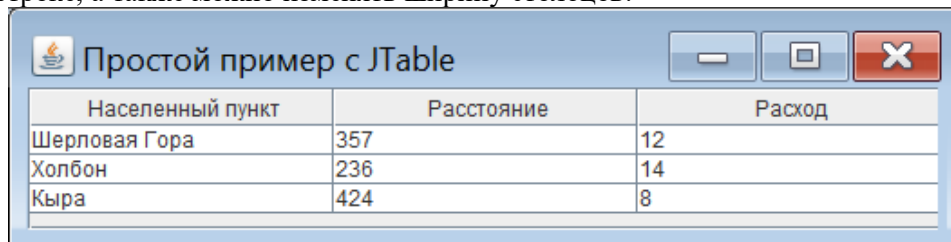
Рассмотрим простой пример создания таблицы на основе массива.

Для начала данные, которые предполагается представить в виде строк таблицы, размещаются в двумерном массиве с именем `array`. Заголовки столбцов заносим в одномерный массив с именем `columnsHeader`. (см. пример1).

```
public class TestForTable
    extends JFrame
{
    private static final long serialVersionUID = 1L;
    //заносим в массив данные, которые затем отобразятся в таблице
    private Object[][] array = new String[][]
    { { "Шерловая гора", "357", "12"},
      { "Холбон", "236", "14" },
      { "Кыра", "424", "8" } };
    // Формируем заголовки столбцов
    private Object[] columnsHeader = new String[] { "Населенный пункт",
    "Расстояние", "Расход" };
    // используем конструктор класса TestForTable
    public TestForTable () {
        super("Заполнение таблицы с помощью массива");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        // Создание простой таблицы на основе массива
        JTable table1 = new JTable(array, columnsHeader);
    }
}
```

Пример.1. Создание простой таблицы.

На этом формирование простой таблицы можно считать законченным. На основе данных массивов создаем новый объект `JTable`. В результате работы приложения получим таблицу (рис.1.), в которой можно выделить одну из строк, выделять несколько строк одновременно, можно получить или изменить данные в строке, а также можно изменять ширину столбцов.



Населенный пункт	Расстояние	Расход
Шерловая Гора	357	12
Холбон	236	14
Кыра	424	8

Рис.1. Простая таблица `JTable` .

Рассмотрим возможности создания таблицы на основе объекта **Vector** (пакет **java.util**). Сначала занесем в первый вектор данные строк таблицы, во второй - названия столбцов таблицы. Для этого организуем цикл и сформируем векторы на основе уже введенных ранее данных массивов (см. пример 2).

```
// Занесем данные для таблицы в объект Vector
    Vector<Vector<String>> data = new Vector<Vector<String>>();
// Занесем заголовки строк
    Vector<String> header = new Vector<String>();
// Сформируем массив данных с помощью цикла
    for (int j = 0; j < array.length; j++) {
        header.add((String)columnsHeader[j]);
        Vector<String> row = new Vector<String>();
        for (int i = 0; i < array[j].length; i++) {
            row.add((String)array[j][i]);
        }
        data.add(row);
    }
```

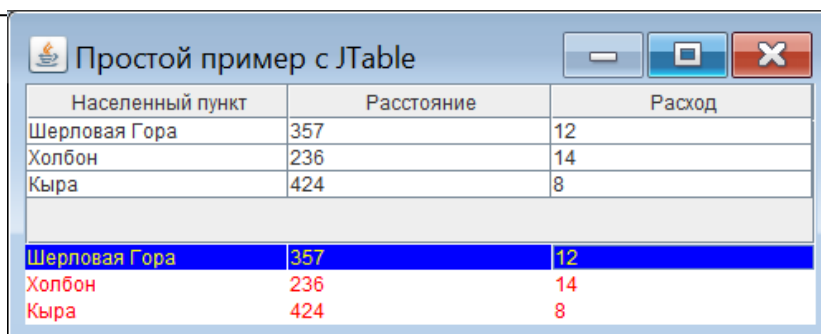
Пример 2. Формирование объекта **Vector** с использованием цикла.

После того, как таблица сформирована, рассмотрим возможности ее форматирования. Для начала разместим обе таблицы в панели **Box** с блочным расположением. Для первой таблицы подключаем возможность прокрутки **JScrollPane**. Затем настроим цвет фона, цвет выделения для текста и фона. Скроем сетку и добавим таблицу на панель. Заканчиваем программу установкой панели вывода с заданными размерами и установкой видимости (см. пример 3).

```
// Размещение таблицы в панели
    Box contents = new Box(BoxLayout.Y_AXIS);
    contents.add(new JScrollPane(table1));
    contents.add(new JScrollPane(table2));
// Для таблицы 2 настроим цвет фона и цвет выделения
    Table2.setForeground(Color.red);
    Table2.setSelectionForeground(Color.yellow);
    Table2.setSelectionBackground(Color.blue);
// Можно скрыть сетку таблицы
    Table2.setShowGrid(false);
    contents.add(table3);
// Выводим окно на экран
    setContentPane(contents);
    setSize(500, 400);
    setVisible(true);
}
}
```

Пример 3. Простая таблица на основе объекта **Vector**.

Для сравнения на рисунке 2 показаны две таблицы, построенные на основе использования массивов, конструктора таблицы **JTable** и объекта **Vector**.



Населенный пункт	Расстояние	Расход
Шерловая Гора	357	12
Холбон	236	14
Кыра	424	8

Рис 2. Простые таблицы на основе объекта Vector.

Таким образом, применение конструкторов **JTable** для организации представления данных в виде простых таблиц полностью отвечает задачам оформления данных в наглядном и удобном для дальнейшей обработки виде.

Литература

1. Шилдт Г. Java 8: Руководство для начинающих, 6-е изд.: Пер. с англ. – М.: ООО «ИД Вильямс», 2015. – 720 с.
2. Шилдт Г. Java 8: Полное руководство, 9-е изд.: Пер. с англ. – М.: ООО «ИД Вильямс», 2015. – 1376 с.
3. Эккель Б. Философия Java. – СПб: Питер, 2016. – 1168 с.
4. Oracle. Integrated Cloud. Applications and Platform Services. [Электронный ресурс]. URL: <http://www.oracle.com/technetwork/java/javase/documentation/index.html>.
5. Библиотека Swing. [Электронный ресурс]. URL: <http://java-online.ru/libs-swing.xhtml>.
6. Мархакшинов А.Л., Шадрина Н.Н. Практикум по программированию на языке Java: практикум. – Улан-Удэ: Издательство БГУ, 2017. – 65 с.
7. Шадрина Н. Н. К вопросу о формировании структуры учебного курса «Программирование на Java». // Информационные системы и технологии в образовании, науке и бизнесе: материалы всероссийской науч.-практ. конф. с междунар. участием (Улан-Удэ, 05 июля 2019 г.). Улан-Удэ: Изд-во Бурят. гос. ун-та, 2019. С. 44.

ORGANIZATION AND DATA PROCESSING IN THE FORM OF JAVA TABLES

Natalya N. Shadrina

Senior Lecturer,

Dorzhi Banzarov Buryat State University

24a Smolina St., Ulan-Ude 670000, Russia

E-mail: shadrinann8@yandex.ru

The article discusses various options for organizing and processing data in the form of tables. The particular attention is paid to the method of organizing simple tables using JTable object constructors. The JTable object is considered, and the related interfaces TableModel, TableColumnModel, SelectionModel which provide to build complex and voluminous tables. The JTable object constructors are also considered, providing to build simple tables with a possibility of various options for presenting data with their placement both in the scroll panel and directly on the form. The examples of component management are provided. The example of building the simple table based on the JTable object constructors is considered in detail.

Keywords. Java programming, JTable object, constructors and interfaces of JTable object