**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение**

**Высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе №4**

По дисциплине «СиАОД»

На тему:

«Реализация стека/дека»

Выполнил: студент группы БВТ1801

Алмамма Башар

Руководитель:

Кутейников Иван Алексеевич

Москва 2020

**Цель работы:**

Используя технологию модульного программирования разработать программу обработки данных, содержащихся в заранее подготовленном файле, в соответствии с индивидуальным заданием. Применить динамическую структуру указанного в задании вида: стек, очередь или дек. Программа должна включать модуль, содержащий набор всех необходимых средств (типов, подпрограмм и т.д.) для решения поставленной задачи.

**Задача (Вариант 1):**Отсортировать строки файла, содержащие названий книг, в алфавитном порядке с использованием двух деков.

**Программа:**

**1.** Классы:

1. Main:  
   Реализует чтение файла,   
   обозначения необходимых деков,   
   прогон по файлу и заполнение дека

Сортировка с помощью 2-х деков  
вывод результата в консоль.

1. Deque:

Поля:

dequeSize: длина дека

dequeArray: дек

top: индекс верхнего элемента

bot: индекс нижнего элемента

Методы:

isEmpty: проверка на пустоту дека

isFull: проверка на полноту дека

addElementTop: добавить элемент сверху в дек

addElementBot: добавить элемент снизу в дек

deleteElementTop: удалить верхний элемент из дека и вернуть его значение

deleteElementBot: удалить нижний элемент из дека и вернуть его значение

getTop: вернуть значение верхнего элемента

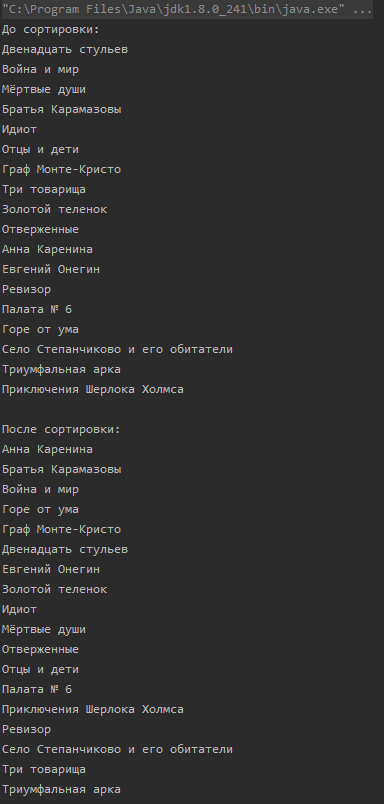
getBot: вернуть значение нижнего элемента

**2.** Код программы:

public class Deque{  
 private int dequeSize;  
 private String[] dequeArray;  
 private int top;  
 private int bot;  
  
 public Deque(int n){  
 this.dequeSize = n;  
 this.dequeArray = new String[dequeSize];  
 this.top = -1;  
 this.bot = 0;  
 }  
  
 public void addElementTop(String element){  
 dequeArray[++top] = element;  
 }  
  
 public void addElementBot(String element){  
 for (int i = top+1; i > 0; i--) {  
 dequeArray[i] = dequeArray[i - 1];  
 }  
 dequeArray[bot] = element;  
 top++;  
 }  
  
 public String deleteElementTop(){  
 return dequeArray[top--];  
 }  
  
 public String deleteElementBot(){  
 String temp = dequeArray[0];  
 for (int i = 0; i < top; i++) {  
 dequeArray[i] = dequeArray[i + 1];  
 }  
 top--;  
 return temp;  
 }  
  
 public String getTop(){  
 return dequeArray[top];  
 }  
  
 public String getBot(){  
 return dequeArray[bot];  
 }  
  
 public boolean isEmpty(){  
 return (top + 1 == bot);  
 }  
  
 public boolean isFull(){  
 return (top == dequeSize - 1);  
 }  
  
}

import java.io.IOException;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
import java.util.List;  
  
public class Main {  
  
 public static void main(String[] args) throws IOException {  
 Path path = Paths.*get*("D:\\Projects\\SiAOD\\Laba4-Java-Stack\\test.txt");  
 List<String> contents = Files.*readAllLines*(path);  
  
 int maxSize = contents.size();  
 Deque deque1 = new Deque(maxSize);  
 Deque deque2 = new Deque(maxSize);  
  
 System.*out*.println("До сортировки:");  
 for (int i = 0; i < contents.size(); i++) {  
 deque1.addElementTop(contents.get(i));  
 System.*out*.println(contents.get(i));  
 }  
  
 while (!deque1.isEmpty()){  
 if(deque2.isEmpty()){  
 deque2.addElementTop(deque1.getTop());  
 deque1.deleteElementTop();  
 }  
 if(deque1.getTop().compareTo(deque2.getTop()) < 0){  
 deque2.addElementTop(deque1.getTop());  
 deque1.deleteElementTop();  
 }  
 else if(deque1.getTop().compareTo(deque2.getBot()) > 0){  
 deque2.addElementBot(deque1.getTop());  
 deque1.deleteElementTop();  
 }  
 else {  
 deque1.addElementBot(deque2.getBot());  
 deque2.deleteElementBot();  
 }  
 }  
  
  
 System.*out*.println();  
 System.*out*.println("После сортировки:");  
 while (!deque2.isEmpty()){  
 System.*out*.println(deque2.deleteElementTop());  
 }  
 }  
}

**3.** Тестирование программы



**Вывод:** в данной лабораторной работе я научился реализовывать деки для улучшенного понимания их работы, а также попробовал решить одну из задач для освоения навыка использования деков в работе.