|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 5 |

**Название:** Исключения, файлы

**Дисциплина:** Языки программирования для работы с большими данными

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | С.В. Астахов | |
|  | (Группа) |  | (Подпись, дата) | | (И.О. Фамилия) |
|  |  |  |  | |  |
| Преподаватель |  |  |  | | П.В. Степанов |
|  |  |  | (Подпись, дата) | | (И.О. Фамилия) |

Москва, 2024

**Цель работы:** освоить базовые принципы работы с исключениями и файлами на языке Java.

**Вариант: 1.**

**Задание 1:** Выполнить задания на основе варианта 1 лабораторной работы 3, контролируя состояние потоков ввода/вывода. При возникновении ошибок, связанных с корректностью выполнения математических операций, генерировать и обрабатывать исключительные ситуации. Предусмотреть обработку исключений, возникающих при нехватке памяти, отсутствии требуемой записи (объекта) в файле, недопустимом значении поля и т.д.

Код решения приведен в листинге 1.

Листинг 1 — реализация решения

|  |
| --- |
| package org.example;  import java.util.Arrays;  import java.util.Scanner;  import static java.lang.Math.\*;  public class Main {  public static void main(String[] args) {  System.out.println("=== Vector class work example ===\n");  Scanner sc = new Scanner(System.in);  int m = 0;  String buf;  boolean retry\_fl = true;  while(retry\_fl) {  try {  System.out.println("Enter number of vectors:");  buf = sc.next();  m = Integer.parseInt(buf);  retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: num format is incorrect!");  }  }  Vector[] vectors = new Vector[m];  for(int i = 0; i < m; i++) {  retry\_fl = true;  int sz = 0;  while(retry\_fl) {  try {  System.out.printf("Enter size at vectors[%d]:\n", i);  buf = sc.next();  sz = Integer.parseInt(buf);  retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: num format is incorrect!");  }  if (sz > 0) {  double[] elems = new double[sz];  for(int j = 0; j < sz; j++) {  boolean inner\_retry\_fl = true;  while(inner\_retry\_fl) {  try {  System.out.printf("Enter elem at vectors[%d][%d]:\n", i, j);  buf = sc.next();  elems[j] = Double.parseDouble(buf);  inner\_retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: num format is incorrect!");  } catch (Exception e) {  System.out.printf("Unknown exception: %s \n", e);  }  }  }  vectors[i] = new Vector(elems);  }  }  }  for(int i = 0; i < m; i++) {  for(int j = i; j < m; j++) {  if (i != j) {  try {  showVectorPair(vectors[i], vectors[j], i, j);  } catch (NullPointerException e) {  System.out.println("Exception: vector is not init!");  } catch (VectorException e) {  System.out.printf("VectorException: %s \n", e);  } catch (Exception e) {  System.out.printf("Unknown exception: %s \n", e);  }  }  }  }  }  static void showVectorPair(Vector v1, Vector v2, int v1\_idx, int v2\_idx) throws Exception {  System.out.printf("=== Vectors %d-th and %d-th ===\n\n", v1\_idx, v2\_idx);  System.out.printf("v%d: %s (len:%7.2f) %nv%d: %s (len:%7.2f)%n%n", v1\_idx, v1, v1.spaceLength(), v2\_idx, v2, v2.spaceLength());  System.out.printf("v%d + v%d: %s%n%n", v1\_idx, v2\_idx, v1.add(v2));  System.out.printf("v%d - v%d: %s%n%n", v1\_idx, v2\_idx, v1.sub(v2));  System.out.printf("v%d \* v%d (scalar): %7.2f%n%n", v1\_idx, v2\_idx, v1.scalar\_mul(v2));  System.out.printf("angle(v%d, v%d)[degrees]: %7.2f%n%n", v1\_idx, v2\_idx, v1.angle(v2));  System.out.printf("v%d++: %s %nv%d--: %s%n%n", v1\_idx, v1.inc(), v1\_idx, v2.dec());  }  }  class Vector {  public double[] data;  public Vector(double[] p\_data) {  this.data = p\_data;  }  public String toString() {  StringBuilder s = new StringBuilder("[ ");  for (double elem : this.data) {  s.append(String.format("%7.2f ", elem));  }  s.append("]");  return String.valueOf(s);  }  public Vector add(Vector v) throws Exception {  Vector result = new Vector(new double[this.data.length]);  if (this.data.length != v.data.length) {  throw new VectorException("Sizes don't match!");  }  for(int i = 0; i < this.data.length; i++) {  result.data[i] = this.data[i] + v.data[i];  }  return result;  }  public Vector sub(Vector v) throws Exception {  Vector result = new Vector(new double[this.data.length]);  if (this.data.length != v.data.length) {  throw new VectorException("Sizes don't match!");  }  for(int i = 0; i < this.data.length; i++) {  result.data[i] = this.data[i] - v.data[i];  }  return result;  }  public double scalar\_mul(Vector v) throws Exception {  double result = 0;  if (this.data.length != v.data.length) {  throw new VectorException("Sizes don't match!");  }  for(int i = 0; i < this.data.length; i++) {  result += this.data[i] \* v.data[i];  }  return result;  }  public Vector inc() {  Vector result = new Vector(new double[this.data.length]);  for (int i = 0; i < this.data.length; i++) {  result.data[i] = this.data[i] + 1;  }  return result;  }  public Vector dec() {  Vector result = new Vector(new double[this.data.length]);  for(int i = 0; i < this.data.length; i++) {  result.data[i] = this.data[i] - 1;  }  return result;  }  public double getByIdx(int i) {  return this.data[i];  }  public void setByIdx(int i, double val) {  this.data[i] = val;  }  public double spaceLength() {  double result = 0.0F;  for (double elem : this.data) {  result += pow(elem, 2);  }  return sqrt(result);  }  public double angle(Vector v) throws Exception {  return Math.toDegrees(  acos(  this.scalar\_mul(v) / (this.spaceLength() \* v.spaceLength())  )  );  }  }  class VectorException extends Exception {  public VectorException(String message) {  super(message);  }  } |

Результат исполнения программы приведен в листинге 2.

Листинг 2 — результат исполнения программы

|  |
| --- |
| === Vector class work example ===  Enter number of vectors:  аа  Exception: num format is incorrect!  Enter number of vectors:  2  Enter size at vectors[0]:  ff  Exception: num format is incorrect!  Enter size at vectors[0]:  2  Enter elem at vectors[0][0]:  1  Enter elem at vectors[0][1]:  1  Enter size at vectors[1]:  2  Enter elem at vectors[1][0]:  2  Enter elem at vectors[1][1]:  4  === Vectors 0-th and 1-th ===  v0: [ 1,00 1,00 ] (len: 1,41)  v1: [ 2,00 4,00 ] (len: 4,47)  v0 + v1: [ 3,00 5,00 ]  v0 - v1: [ -1,00 -3,00 ]  v0 \* v1 (scalar): 6,00  angle(v0, v1)[degrees]: 18,43  v0++: [ 2,00 2,00 ]  v0--: [ 1,00 3,00 ] |

**Задание 2:** см. задание 1.

Код решения приведен в листинге 3.

Листинг 3 — реализация решения

|  |
| --- |
| package org.example;  /\*  2. Определить класс Вектор размерности n. Определить несколько конструкторов.  Реализовать методы для вычисления модуля вектора, скалярного произведения, сложения,  вычитания, умножения на константу.  Объявить массив объектов. Написать метод, который для заданной пары векторов будет определять,  являются ли они коллинеарными или ортогональными.  \*/  import java.util.ArrayList;  import java.util.Arrays;  import java.util.Scanner;  import static java.lang.Math.\*;  public class Main {  public static void main(String[] args) {  System.out.println("=== Vector class work example ===\n");  Scanner sc = new Scanner(System.in);  int m = 0;  String buf;  boolean retry\_fl = true;  while(retry\_fl) {  try {  System.out.println("Enter number of vectors:");  buf = sc.next();  m = Integer.parseInt(buf);  retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: num format is incorrect!");  }  }  Vector[] vectors = new Vector[m];  for(int i = 0; i < m; i++) {  retry\_fl = true;  int sz = 0;  while(retry\_fl) {  try {  System.out.printf("Enter size at vectors[%d]:\n", i);  buf = sc.next();  sz = Integer.parseInt(buf);  retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: num format is incorrect!");  }  if (sz > 0) {  double[] elems = new double[sz];  for(int j = 0; j < sz; j++) {  boolean inner\_retry\_fl = true;  while(inner\_retry\_fl) {  try {  System.out.printf("Enter elem at vectors[%d][%d]:\n", i, j);  buf = sc.next();  elems[j] = Double.parseDouble(buf);  inner\_retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: num format is incorrect!");  }  }  }  vectors[i] = new Vector(elems);  }  }  }  for(int i = 0; i < m; i++) {  for(int j = i; j < m; j++) {  if (i != j) {  try {  showVectorPair(vectors[i], vectors[j], i, j);  } catch (NullPointerException e) {  System.out.println("Exception: vector is not init!");  } catch (VectorException e) {  System.out.printf("VectorException: %s \n", e);  }  catch (Exception e) {  System.out.printf("Unknown exception: %s \n", e);  }  }  }  }  }  static void showVectorPair(Vector v1, Vector v2, int v1\_idx, int v2\_idx) throws Exception {  System.out.printf("=== Vectors %d-th and %d-th ===\n\n", v1\_idx, v2\_idx);  System.out.printf("v%d: %s (len:%7.2f) %nv%d: %s (len:%7.2f)%n%n", v1\_idx, v1, v1.module(), v2\_idx, v2, v2.module());  System.out.printf("v%d + v%d: %s%n%n", v1\_idx, v2\_idx, v1.add(v2));  System.out.printf("v%d - v%d: %s%n%n", v1\_idx, v2\_idx, v1.sub(v2));  System.out.printf("v%d \* v%d (scalar): %7.2f%n%n", v1\_idx, v2\_idx, v1.scalar\_mul(v2));  System.out.printf("v%d and v%d are orthogonal: %b%n%n", v1\_idx, v2\_idx, v1.isOrtogonal(v2));  System.out.printf("v%d and v%d are collinear: %b%n%n", v1\_idx, v2\_idx, v1.isCollinear(v2));  }  }  class Vector {  public double[] data;  public Vector(double[] p\_data) {  this.data = p\_data;  }  public Vector(ArrayList<Double> p\_data) {  this.data = new double[p\_data.size()];  for (int i = 0; i < p\_data.size(); i++) {  this.data[i] = p\_data.get(i);  }  }  public Vector(Vector p\_data) {  this.data = p\_data.data.clone();  }  public Vector(int k) {  this.data = new double[k];  for (int i = 0; i < k; i++) {  this.data[i] = 0.0;  }  }  public String toString() {  StringBuilder s = new StringBuilder("[ ");  for (double elem : this.data) {  s.append(String.format("%7.2f ", elem));  }  s.append("]");  return String.valueOf(s);  }  public Vector add(Vector v) throws Exception {  Vector result = new Vector(new double[this.data.length]);  if (this.data.length != v.data.length) {  throw new VectorException("Sizes don't match!");  }  for(int i = 0; i < this.data.length; i++) {  result.data[i] = this.data[i] + v.data[i];  }  return result;  }  public Vector sub(Vector v) throws Exception {  Vector result = new Vector(new double[this.data.length]);  if (this.data.length != v.data.length) {  throw new VectorException("Sizes don't match!");  }  for(int i = 0; i < this.data.length; i++) {  result.data[i] = this.data[i] - v.data[i];  }  return result;  }  public double scalar\_mul(Vector v) throws Exception {  double result = 0;  if (this.data.length != v.data.length) {  throw new VectorException("Sizes don't match!");  }  for(int i = 0; i < this.data.length; i++) {  result += this.data[i] \* v.data[i];  }  return result;  }  public Vector const\_mul(double k) throws Exception {  Vector result = new Vector(new double[this.data.length]);  for(int i = 0; i < this.data.length; i++) {  result.data[i] = this.data[i] \* k;  }  return result;  }  public double module() {  double result = 0.0F;  for (double elem : this.data) {  result += pow(elem, 2);  }  return sqrt(result);  }  public boolean isCollinear(Vector v) throws Exception {  double epsilon = 0.001;  boolean result = true;  double k = 0;  boolean k\_init = false;  if (this.data.length != v.data.length) {  throw new VectorException("Sizes don't match!");  }  for(int i = 0; i < this.data.length; i++) {  if(abs(this.data[i]) > epsilon && abs(v.data[i]) > epsilon && !k\_init) {  k = this.data[i] / v.data[i];  k\_init = true;  } else if (abs(this.data[i]) > epsilon && abs(v.data[i]) > epsilon && k\_init) {  if(abs(this.data[i] / v.data[i] - k) > epsilon) {  result = false;  }  }  if(abs(this.data[i]) > epsilon ^ abs(v.data[i]) > epsilon) {  result = false;  }  }  return result;  }  public boolean isOrtogonal(Vector v) throws Exception {  double epsilon = 0.001;  return abs(this.scalar\_mul(v)) < epsilon;  }  }  class VectorException extends Exception {  public VectorException(String message) {  super(message);  }  } |

Результат исполнения программы приведен в листинге 4.

Листинг 4 — результат исполнения программы

|  |
| --- |
| === Vector class work example ===  Enter number of vectors:  2  Enter size at vectors[0]:  аа  Exception: num format is incorrect!  Enter size at vectors[0]:  2  Enter elem at vectors[0][0]:  1  Enter elem at vectors[0][1]:  0  Enter size at vectors[1]:  2  Enter elem at vectors[1][0]:  0  Enter elem at vectors[1][1]:  1  === Vectors 0-th and 1-th ===  v0: [ 1,00 0,00 ] (len: 1,00)  v1: [ 0,00 1,00 ] (len: 1,00)  v0 + v1: [ 1,00 1,00 ]  v0 - v1: [ 1,00 -1,00 ]  v0 \* v1 (scalar): 0,00  v0 and v1 are orthogonal: true  v0 and v1 are collinear: false |

**Задание 3:** Выполнить задания из варианта 2 лабораторной работы 3, реализуя собственные обработчики исключений и исключения ввода/вывода.

Код решения приведен в листинге 5.

Листинг 5 — реализация решения

|  |
| --- |
| package org.example;  /\*  Создать классы, спецификации которых приведены ниже.  Определить конструкторы и методы setТип(), getТип(), toString().  Определить дополнительно методы в классе, создающем массив объектов.  Задать критерий выбора данных и вывести эти данные на консоль.  1. Student: id, Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс, Группа.  Создать массив объектов. Вывести:  a) список студентов заданного факультета;  b) списки студентов для каждого факультета и курса;  c) список студентов, родившихся после заданного года;  d) список учебной группы.  \*/  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.\*;  public class Main {  public static void main(String[] args) throws ParseException {  System.out.println("=== StudentCollection class example work ===");  SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");  Scanner sc = new Scanner(System.in);  int m = 0;  String buf;  boolean retry\_fl = true;  while(retry\_fl) {  try {  System.out.println("Enter number of students:");  buf = sc.next();  m = Integer.parseInt(buf);  retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: num format is incorrect!");  }  }  Student[] students = new Student[m];  for(int i = 0; i < m; i++) {  boolean global\_retry\_fl = true;  while(global\_retry\_fl) {  System.out.printf("Enter name of student[%d]:\n", i);  String name = sc.next();  System.out.printf("Enter surnmame of student[%d]:\n", i);  String surname = sc.next();  System.out.printf("Enter thirdname of student[%d]:\n", i);  String thirdname = sc.next();  Date birthdate = dateFormat.parse("1900-01-01");  retry\_fl = true;  while (retry\_fl) {  try {  System.out.printf("Enter birthdate of student[%d]:\n", i);  buf = sc.next();  birthdate = dateFormat.parse(buf);  retry\_fl = false;  } catch (ParseException e) {  System.out.println("Exception: date format is incorrect!");  }  }  System.out.printf("Enter address of student[%d]:\n", i);  String address = sc.next();  System.out.printf("Enter phone of student[%d]:\n", i);  String phone = sc.next();  System.out.printf("Enter faculty of student[%d]:\n", i);  String faculty = sc.next();  int course = 1;  int group = 1;  retry\_fl = true;  while (retry\_fl) {  try {  System.out.printf("Enter course of student[%d]:\n", i);  buf = sc.next();  course = Integer.parseInt(buf);  retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: date format is incorrect!");  }  }  retry\_fl = true;  while (retry\_fl) {  try {  System.out.printf("Enter group of student[%d]:\n", i);  buf = sc.next();  group = Integer.parseInt(buf);  retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: date format is incorrect!");  }  }  try {  students[i] = new Student(i, name, surname, thirdname, birthdate, address, phone, faculty, course, group);  global\_retry\_fl = false;  } catch (StudentException e) {  System.out.printf("Student Exception: %s \n", e);  }  }  }  StudentCollection st\_collection = new StudentCollection(students);  System.out.println("=== Filter by faculty (SM) ===");  try {  System.out.println(st\_collection.filterByFaculty("SM"));  } catch (StudentException e) {  System.out.printf("Student Exception: %s \n", e);  }  System.out.println("=== Filter by faculty (guhihkj) ===");  try {  System.out.println(st\_collection.filterByFaculty("guhihkj"));  } catch (StudentException e) {  System.out.printf("Student Exception: %s \n", e);  }  System.out.println("=== Filter by birthyear (1999) ===");  try {  System.out.println(st\_collection.filterByBirthYear(1999));  } catch (StudentException e) {  System.out.printf("Student Exception: %s \n", e);  }  System.out.println("=== Filter by birthyear (-1) ===");  try {  System.out.println(st\_collection.filterByBirthYear(1999));  } catch (StudentException e) {  System.out.printf("Student Exception: %s \n", e);  }  System.out.println("=== Filter by group (ICS, 5, 2) ===");  try {  System.out.println(st\_collection.filterByGroup("ICS", 5, 2));  } catch (StudentException e) {  System.out.printf("Student Exception: %s \n", e);  }  System.out.println("=== Filter by group (biuio, 5, 2) ===");  try {  System.out.println(st\_collection.filterByGroup("biuio", 5, 2));  } catch (StudentException e) {  System.out.printf("Student Exception: %s \n", e);  }  System.out.println("=== All by faculty and course ===");  System.out.println(st\_collection.toStringByFacultyAndCourse());  }  }  class StudentCollection {  private Student[] data;  public StudentCollection(Student[] data) {  this.data = data;  }  public String toString() {  String result = "";  for(Student elem: this.data) {  result = String.format("%s%n%n%s", result, elem.toString());  }  return result;  }  public StudentCollection filterByFaculty(String p\_faculty) throws StudentException {  if (!Student.faculties.contains(p\_faculty)){  throw new StudentException("invalid faculty!");  }  Student[] result;  Student[] filtered = Arrays.stream(this.data)  .filter(x -> Objects.equals(x.getFaculty(), p\_faculty))  .toArray(Student[]::new);  return new StudentCollection(filtered);  }  public Student[] getData() {  return this.data;  }  public Map<String, Map<Integer, StudentCollection>> getByFacultyAndCourse() {  Map<String, Map<Integer, StudentCollection>> result = new HashMap<>();  for (Student elem : this.data) {  Map<Integer, StudentCollection> default\_inner\_map = new HashMap<>();  Student[] tmp\_data = result  .getOrDefault(  elem.getFaculty(),  default\_inner\_map  )  .getOrDefault(  elem.getCourse(),  new StudentCollection(new Student[]{})  )  .getData();  Student[] new\_data = Arrays.copyOf(tmp\_data, tmp\_data.length + 1);  new\_data[tmp\_data.length] = elem;  Map<Integer, StudentCollection> tmp\_inner\_map;  tmp\_inner\_map = result.getOrDefault(  elem.getFaculty(),  default\_inner\_map  );  tmp\_inner\_map.put(elem.getCourse(), new StudentCollection(new\_data));  result.put(elem.getFaculty(), tmp\_inner\_map);  }  return result;  }  public String toStringByFacultyAndCourse() {  String result = "";  Map<String, Map<Integer, StudentCollection>> tmp = this.getByFacultyAndCourse();  for(String faculty\_key : tmp.keySet()){  for(int course\_key : tmp.get(faculty\_key).keySet()){  result = String.format("%s%n%n=== faculty: %s course: %d ===%n%n%s",  result, faculty\_key, course\_key, tmp.get(faculty\_key).get(course\_key).toString());  }  }  return result;  }  public StudentCollection filterByBirthYear(int year) throws StudentException {  Student[] result;  Date date;  if(year < 0) {  throw new StudentException("Invalid date!");  }  SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy");  try {  date = dateFormat.parse(String.format("%d", year + 1)); // Because we need end of year  } catch (ParseException e) {  throw new StudentException("Invalid date!");  }  Student[] filtered = Arrays.stream(this.data)  .filter(x -> x.getBirthdate().after(date))  .toArray(Student[]::new);  return new StudentCollection(filtered);  }  public StudentCollection filterByGroup(String faculty, int course, int group) throws StudentException {  Student[] result;  if (course < 1 || course > 6){  throw new StudentException("invalid course!");  }  if (group < 1){  throw new StudentException("invalid group!");  }  if (!Student.faculties.contains(faculty)){  throw new StudentException("invalid faculty!");  }  Student[] filtered = Arrays.stream(this.data)  .filter(x -> x.getFaculty().equals(faculty)  && x.getCourse() == course  && x.getGroup() == group  )  .toArray(Student[]::new);  return new StudentCollection(filtered);  }  }  class Student {  private int id;  private String name;  private String surname;  private String thirdname;  private Date birthdate;  private String address;  private String phone;  private String faculty;  private int course;  private int group;  public static Set<String> faculties = Set.of("ICS", "SM", "MT");  public Student(int id, String name, String surname, String thirdname, Date birthdate, String address, String phone, String faculty, int course, int group) throws StudentException {  this.id = id;  this.name = name;  this.surname = surname;  this.thirdname = thirdname;  this.birthdate = birthdate;  this.address = address;  this.phone = phone;  if (!faculties.contains(faculty)){  throw new StudentException("Ivalid faculty!");  }  this.faculty = faculty;  if (course < 1 || course > 6){  throw new StudentException("Ivalid course!");  }  this.course = course;  if (group < 1){  throw new StudentException("Ivalid group!");  }  this.group = group;  }  public String toString() {  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  return String.format("Student(id=%d): %s %s %s (%s course: %d groud: %d)\n", this.id, this.name, this.surname, this.thirdname, this.faculty, this.course, this.group) +  String.format("Birthdate: %s Phone: %s\n", sdf.format(this.birthdate), this.phone) +  String.format("Address: %s\n", this.address);  }  public int getId() {  return id;  }  public String getName() {  return name;  }  public String getSurname() {  return surname;  }  public String getThirdname() {  return thirdname;  }  public Date getBirthdate() {  return birthdate;  }  public String getAddress() {  return address;  }  public String getPhone() {  return phone;  }  public String getFaculty() {  return faculty;  }  public int getCourse() {  return course;  }  public int getGroup() {  return group;  }  public void setId(int id) {  this.id = id;  }  public void setName(String name) {  this.name = name;  }  public void setSurname(String surname) {  this.surname = surname;  }  public void setThirdname(String thirdname) {  this.thirdname = thirdname;  }  public void setBirthdate(Date birthdate) {  this.birthdate = birthdate;  }  public void setAddress(String address) {  this.address = address;  }  public void setPhone(String phone) {  this.phone = phone;  }  public void setFaculty(String faculty) {  this.faculty = faculty;  }  public void setCourse(int course) {  this.course = course;  }  public void setGroup(int group) {  this.group = group;  }  }  class StudentException extends Exception {  public StudentException(String message) {  super(message);  }  } |

Результат исполнения программы приведен в листинге 6.

Листинг 6 — результат исполнения программы

|  |
| --- |
| === StudentCollection class example work ===  Enter number of students:  аа  Exception: num format is incorrect!  Enter number of students:  2  Enter name of student[0]:  Sergey  Enter surnmame of student[0]:  X  Enter thirdname of student[0]:  X  Enter birthdate of student[0]:  2001-07-16  Enter address of student[0]:  Msk  Enter phone of student[0]:  8-800-...  Enter faculty of student[0]:  ICS  Enter course of student[0]:  ff  Exception: date format is incorrect!  Enter course of student[0]:  4  Enter group of student[0]:  2  Enter name of student[1]:  Max  Enter surnmame of student[1]:  X  Enter thirdname of student[1]:  X  Enter birthdate of student[1]:  2002-01-01  Enter address of student[1]:  Ekb  Enter phone of student[1]:  8-900-...  Enter faculty of student[1]:  ICS  Enter course of student[1]:  3  Enter group of student[1]:  1  === Filter by faculty (SM) ===  === Filter by faculty (guhihkj) ===  Student Exception: org.example.StudentException: invalid faculty!  === Filter by birthyear (1999) ===  Student(id=0): Sergey X X (ICS course: 4 groud: 2)  Birthdate: 2001-07-16 Phone: 8-800-...  Address: Msk  Student(id=1): Max X X (ICS course: 3 groud: 1)  Birthdate: 2002-01-01 Phone: 8-900-...  Address: Ekb  === Filter by birthyear (-1) ===  Student(id=0): Sergey X X (ICS course: 4 groud: 2)  Birthdate: 2001-07-16 Phone: 8-800-...  Address: Msk  Student(id=1): Max X X (ICS course: 3 groud: 1)  Birthdate: 2002-01-01 Phone: 8-900-...  Address: Ekb  === Filter by group (ICS, 5, 2) ===  === Filter by group (biuio, 5, 2) ===  Student Exception: org.example.StudentException: invalid faculty!  === All by faculty and course ===  === faculty: ICS course: 3 ===  Student(id=1): Max X X (ICS course: 3 groud: 1)  Birthdate: 2002-01-01 Phone: 8-900-...  Address: Ekb  === faculty: ICS course: 4 ===  Student(id=0): Sergey X X (ICS course: 4 groud: 2)  Birthdate: 2001-07-16 Phone: 8-800-...  Address: Msk |

**Задание 4:** см. задание 3.

Код решения приведен в листинге 7.

Листинг 7 — реализация решения

|  |
| --- |
| package org.example;  /\*  Создать классы, спецификации которых приведены ниже.  Определить конструкторы и методы setТип(), getТип(), toString().  Определить дополнительно методы в классе, создающем массив объектов.  Задать критерий выбора данных и вывести эти данные на консоль.  2. Customer: id, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счета.  Создать массив объектов. Вывести:  a) список покупателей в алфавитном порядке;  b) список покупателей, у которых номер кредитной карточки находится в заданном интервале.  \*/  import java.util.\*;  public class Main {  public static void main(String[] args) {  System.out.println("=== Customer class work example ===\n\n");  Scanner sc = new Scanner(System.in);  int m = 0;  String buf;  boolean retry\_fl = true;  while(retry\_fl) {  try {  System.out.println("Enter number of customers:");  buf = sc.next();  m = Integer.parseInt(buf);  retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: num format is incorrect!");  }  }  Customer[] customers = new Customer[m];  for(int i = 0; i < m; i++) {  System.out.printf("Enter name of customer[%d]:\n", i);  String name = sc.next();  System.out.printf("Enter surnmame of customer[%d]:\n", i);  String surname = sc.next();  System.out.printf("Enter thirdname of customer[%d]:\n", i);  String thirdname = sc.next();  System.out.printf("Enter address of customer[%d]:\n", i);  String address = sc.next();  String card\_num = "";  retry\_fl = true;  while(retry\_fl) {  try {  System.out.printf("Enter number of card of customer[%d]:\n", i);  card\_num = sc.next();  Long.parseLong(card\_num.replace("-", ""));  retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: num format is incorrect!");  }  }  String account\_num = "";  retry\_fl = true;  while(retry\_fl) {  try {  System.out.printf("Enter number of account of customer[%d]:\n", i);  account\_num = sc.next();  Long.parseLong(account\_num.replace("-", ""));  retry\_fl = false;  } catch (NumberFormatException e) {  System.out.println("Exception: num format is incorrect!");  }  }  customers[i] = new Customer(name, surname, thirdname, address, card\_num, account\_num);  }  CustomerCollection cs\_collection = new CustomerCollection(customers);  System.out.println("=== Customers sorted by alphabet ===\n\n");  System.out.println(cs\_collection.orderByAlphabet());  System.out.println("=== Customers filtered by cardnums(1111-2222-0001, 1111-2222-0003) ===\n\n");  try {  System.out.println(cs\_collection.filterByCardnums("1111-2222-0001", "1111-2222-0003"));  } catch (CustomerException e) {  System.out.printf("Customer exception: %s\n", e);  }  System.out.println("=== Customers filtered by cardnums(1111-ffff-0001, 1111-2222-0003) ===\n\n");  try {  System.out.println(cs\_collection.filterByCardnums("1111-ffff-0001", "1111-2222-0003"));  } catch (CustomerException e) {  System.out.printf("Customer exception: %s\n", e);  }  }  }  class CustomerCollection {  Customer[] data;  public CustomerCollection(Customer[] p\_data) {  this.data = p\_data;  }  public String toString() {  String result = "";  for(Customer elem: data){  result = String.format("%s %n%n%s", result, elem);  }  return result;  }  public CustomerCollection orderByAlphabet() {  List<Customer> data\_list = Arrays.asList(this.data);  data\_list.sort(Comparator.comparing(x -> x.getFio()));  return new CustomerCollection(data\_list.toArray(new Customer[0]));  }  public CustomerCollection filterByCardnums(String num1, String num2) throws CustomerException {  try {  Long.parseLong(num1.replace("-", ""));  Long.parseLong(num2.replace("-", ""));  } catch (NumberFormatException e) {  throw new CustomerException("Invalid card number!");  }  Customer[] filtered = Arrays.stream(this.data)  .filter(x ->  Long.parseLong(x.getCardNumber().replace("-", "")) >=  Long.parseLong(num1.replace("-", ""))  && Long.parseLong(x.getCardNumber().replace("-", "")) <=  Long.parseLong(num2.replace("-", ""))  )  .toArray(Customer[]::new);  return new CustomerCollection(filtered);  }  }  class Customer {  static int idCnt = 0;  int id;  String surname;  String name;  String thirdname;  String address;  String card\_number;  String account\_number;  public Customer(String surname, String name, String thirdname, String address, String card\_number, String account\_number) {  this.id = idCnt;  this.surname = surname;  this.name = name;  this.thirdname = thirdname;  this.address = address;  this.card\_number = card\_number;  this.account\_number = account\_number;  idCnt++;  }  public String toString() {  return String.format("Client(id=%d) %s %s %s%n", this.id, this.surname, this.name, this.thirdname) +  String.format("Address: %s%n", this.address) +  String.format("Card: %s Account: %s%n", this.card\_number, this.account\_number);  }  public String getFio() {  return String.format("%s %s %s", this.surname, this.name, this.thirdname);  }  public int getId() {  return id;  }  public String getSurname() {  return surname;  }  public String getName() {  return name;  }  public String getThirdname() {  return thirdname;  }  public String getAddress() {  return address;  }  public String getCardNumber() {  return card\_number;  }  public String getAccountNumber() {  return account\_number;  }  public void setId(int id) {  this.id = id;  }  public void setSurname(String surname) {  this.surname = surname;  }  public void setName(String name) {  this.name = name;  }  public void setThirdname(String thirdname) {  this.thirdname = thirdname;  }  public void setAddress(String address) {  this.address = address;  }  public void setCard\_number(String card\_number) {  this.card\_number = card\_number;  }  public void setAccount\_number(String account\_number) {  this.account\_number = account\_number;  }  }  class CustomerException extends Exception {  public CustomerException(String message) {  super(message);  }  } |

Результат исполнения программы приведен в листинге 8.

Листинг 8 — результат исполнения программы

|  |
| --- |
| === Customer class work example ===  Enter number of customers:  2  Enter name of customer[0]:  Sergey  Enter surnmame of customer[0]:  X  Enter thirdname of customer[0]:  X  Enter address of customer[0]:  Msk  Enter number of card of customer[0]:  1111-2222-0001  Enter number of account of customer[0]:  1111-2222-0001  Enter name of customer[1]:  Max  Enter surnmame of customer[1]:  X  Enter thirdname of customer[1]:  X  Enter address of customer[1]:  Ekb  Enter number of card of customer[1]:  1111-1111-2222  Enter number of account of customer[1]:  1111-1111-2222  === Customers sorted by alphabet ===    Client(id=1) Max X X  Address: Ekb  Card: 1111-1111-2222 Account: 1111-1111-2222    Client(id=0) Sergey X X  Address: Msk  Card: 1111-2222-0001 Account: 1111-2222-0001  === Customers filtered by cardnums(1111-2222-0001, 1111-2222-0003) ===    Client(id=0) Sergey X X  Address: Msk  Card: 1111-2222-0001 Account: 1111-2222-0001  === Customers filtered by cardnums(1111-ffff-0001, 1111-2222-0003) ===  Customer exception: org.example.CustomerException: Invalid card number! |

**Задание 5:** В каждой строке найти и удалить заданную подстроку.

Код решения приведен в листинге 9.

Листинг 9 — реализация решения

|  |
| --- |
| package org.example;  import java.io.\*;  import java.util.\*;  public class Main {  public static void main(String[] args) throws IOException {  System.out.println("=== File class work example ===\n\n");  System.out.println("Enter input file name (block.txt): ");  Scanner scanner = new Scanner(System.in);  String in\_filename = scanner.nextLine();  if(Objects.equals(in\_filename, "")) {  in\_filename = "block.txt";  System.out.println("using block.txt");  }  System.out.println("Enter output file name (out.txt): ");  String out\_filename = scanner.nextLine();  if(Objects.equals(out\_filename, "")) {  out\_filename = "out.txt";  System.out.println("using out.txt");  }  File out\_file = new File(out\_filename);  out\_file.createNewFile();  System.out.println("Enter substring to delete:");  String substring = scanner.nextLine();  BufferedReader reader = new BufferedReader(new FileReader(in\_filename));  BufferedWriter writer = new BufferedWriter(new FileWriter(out\_filename));  String currentLine;  System.out.println("File compare:");  while ((currentLine = reader.readLine()) != null) {  System.out.printf("%s\n", currentLine);  System.out.printf("> %s\n", currentLine.replace(substring, ""));  writer.write(currentLine.replace(substring, "") + System.lineSeparator());  }  writer.close();  reader.close();  }  } |

Результат исполнения программы приведен в листинге 10.

Листинг 10 — результат исполнения программы

|  |
| --- |
| === File class work example ===  Enter input file name (block.txt):  using block.txt  Enter output file name (out.txt):  using out.txt  Enter substring to delete:  вин  File compare:  По вечерам над ресторанами  > По вечерам над ресторанами  Горячий воздух дик и глух,  > Горячий воздух дик и глух,  И правит окриками пьяными  > И правит окриками пьяными  Весенний и тлетворный дух.  > Весенний и тлетворный дух.  >  Вдали над пылью переулочной,  > Вдали над пылью переулочной,  Над скукой загородных дач,  > Над скукой загородных дач,  Чуть золотится крендель булочной,  > Чуть золотится крендель булочной,  И раздается детский плач.  > И раздается детский плач.  >  И каждый вечер, за шлагбаумами,  > И каждый вечер, за шлагбаумами,  Заламывая котелки,  > Заламывая котелки,  Среди канав гуляют с дамами  > Среди канав гуляют с дамами  Испытанные остряки.  > Испытанные остряки.  >  Над озером скрипят уключины  > Над озером скрипят уключины  И раздается женский визг,  > И раздается женский визг,  А в небе, ко всему приученный  > А в небе, ко всему приученный  Бесмысленно кривится диск.  > Бесмысленно кривится диск.  >  И каждый вечер друг единственный  > И каждый вечер друг единственный  В моем стакане отражен  > В моем стакане отражен  И влагой терпкой и таинственной  > И влагой терпкой и таинственной  Как я, смирен и оглушен.  > Как я, смирен и оглушен.  >  А рядом у соседних столиков  > А рядом у соседних столиков  Лакеи сонные торчат,  > Лакеи сонные торчат,  И пьяницы с глазами кроликов  > И пьяницы с глазами кроликов  «In vino veritas!»1 кричат.  > «In vino veritas!»1 кричат.  >  И каждый вечер, в час назначенный  > И каждый вечер, в час назначенный  (Иль это только снится мне?),  > (Иль это только снится мне?),  Девичий стан, шелками схваченный,  > Девичий стан, шелками схваченный,  В туманном движется окне.  > В туманном движется окне.  >  И медленно, пройдя меж пьяными,  > И медленно, пройдя меж пьяными,  Всегда без спутников, одна  > Всегда без спутников, одна  Дыша духами и туманами,  > Дыша духами и туманами,  Она садится у окна.  > Она садится у окна.  >  И веют древними поверьями  > И веют древними поверьями  Ее упругие шелка,  > Ее упругие шелка,  И шляпа с траурными перьями,  > И шляпа с траурными перьями,  И в кольцах узкая рука.  > И в кольцах узкая рука.  >  И странной близостью закованный,  > И странной близостью закованный,  Смотрю за темную вуаль,  > Смотрю за темную вуаль,  И вижу берег очарованный  > И вижу берег очарованный  И очарованную даль.  > И очарованную даль.  >  Глухие тайны мне поручены,  > Глухие тайны мне поручены,  Мне чье-то солнце вручено,  > Мне чье-то солнце вручено,  И все души моей излучины  > И все души моей излучины  Пронзило терпкое вино.  > Пронзило терпкое о.  >  И перья страуса склоненные  > И перья страуса склоненные  В моем качаются мозгу,  > В моем качаются мозгу,  И очи синие бездонные  > И очи синие бездонные  Цветут на дальнем берегу.  > Цветут на дальнем берегу.  >  В моей душе лежит сокровище,  > В моей душе лежит сокровище,  И ключ поручен только мне!  > И ключ поручен только мне!  Ты право, пьяное чудовище!  > Ты право, пьяное чудовище!  Я знаю: истина в вине.  > Я знаю: истина в е. |

**Задание 6:** В каждой строке стихотворения Александра Блока найти и заменить заданную подстроку на подстроку иной длины.

Код решения приведен в листинге 11.

Листинг 11 — реализация решения

|  |
| --- |
| package org.example;  import java.io.\*;  import java.util.\*;  public class Main {  public static void main(String[] args) throws IOException {  System.out.println("=== File class work example ===\n\n");  System.out.println("Enter input file name (block.txt): ");  Scanner scanner = new Scanner(System.in);  String in\_filename = scanner.nextLine();  if(Objects.equals(in\_filename, "")) {  in\_filename = "block.txt";  System.out.println("using block.txt");  }  System.out.println("Enter output file name (out.txt): ");  String out\_filename = scanner.nextLine();  if(Objects.equals(out\_filename, "")) {  out\_filename = "out.txt";  System.out.println("using out.txt");  }  File out\_file = new File(out\_filename);  out\_file.createNewFile();  System.out.println("Enter substring to replace (from):");  String substring\_from = scanner.nextLine();  System.out.println("Enter substring to replace (to):");  String substring\_to = scanner.nextLine();  BufferedReader reader = new BufferedReader(new FileReader(in\_filename));  BufferedWriter writer = new BufferedWriter(new FileWriter(out\_filename));  String currentLine;  System.out.println("File compare:");  while ((currentLine = reader.readLine()) != null) {  System.out.printf("%s\n", currentLine);  System.out.printf("> %s\n", currentLine.replace(substring\_from, substring\_to));  writer.write(currentLine.replace(substring\_from, substring\_to));  }  writer.close();  reader.close();  }  } |

Результат исполнения программы приведен в листинге 12.

Листинг 12 — результат исполнения программы

|  |
| --- |
| === File class work example ===  Enter input file name (block.txt):  using block.txt  Enter output file name (out.txt):  using out.txt  Enter substring to replace (from):  вин  Enter substring to replace (to):  winn  File compare:  По вечерам над ресторанами  > По вечерам над ресторанами  Горячий воздух дик и глух,  > Горячий воздух дик и глух,  И правит окриками пьяными  > И правит окриками пьяными  Весенний и тлетворный дух.  > Весенний и тлетворный дух.  >  Вдали над пылью переулочной,  > Вдали над пылью переулочной,  Над скукой загородных дач,  > Над скукой загородных дач,  Чуть золотится крендель булочной,  > Чуть золотится крендель булочной,  И раздается детский плач.  > И раздается детский плач.  >  И каждый вечер, за шлагбаумами,  > И каждый вечер, за шлагбаумами,  Заламывая котелки,  > Заламывая котелки,  Среди канав гуляют с дамами  > Среди канав гуляют с дамами  Испытанные остряки.  > Испытанные остряки.  >  Над озером скрипят уключины  > Над озером скрипят уключины  И раздается женский визг,  > И раздается женский визг,  А в небе, ко всему приученный  > А в небе, ко всему приученный  Бесмысленно кривится диск.  > Бесмысленно кривится диск.  >  И каждый вечер друг единственный  > И каждый вечер друг единственный  В моем стакане отражен  > В моем стакане отражен  И влагой терпкой и таинственной  > И влагой терпкой и таинственной  Как я, смирен и оглушен.  > Как я, смирен и оглушен.  >  А рядом у соседних столиков  > А рядом у соседних столиков  Лакеи сонные торчат,  > Лакеи сонные торчат,  И пьяницы с глазами кроликов  > И пьяницы с глазами кроликов  «In vino veritas!»1 кричат.  > «In vino veritas!»1 кричат.  >  И каждый вечер, в час назначенный  > И каждый вечер, в час назначенный  (Иль это только снится мне?),  > (Иль это только снится мне?),  Девичий стан, шелками схваченный,  > Девичий стан, шелками схваченный,  В туманном движется окне.  > В туманном движется окне.  >  И медленно, пройдя меж пьяными,  > И медленно, пройдя меж пьяными,  Всегда без спутников, одна  > Всегда без спутников, одна  Дыша духами и туманами,  > Дыша духами и туманами,  Она садится у окна.  > Она садится у окна.  >  И веют древними поверьями  > И веют древними поверьями  Ее упругие шелка,  > Ее упругие шелка,  И шляпа с траурными перьями,  > И шляпа с траурными перьями,  И в кольцах узкая рука.  > И в кольцах узкая рука.  >  И странной близостью закованный,  > И странной близостью закованный,  Смотрю за темную вуаль,  > Смотрю за темную вуаль,  И вижу берег очарованный  > И вижу берег очарованный  И очарованную даль.  > И очарованную даль.  >  Глухие тайны мне поручены,  > Глухие тайны мне поручены,  Мне чье-то солнце вручено,  > Мне чье-то солнце вручено,  И все души моей излучины  > И все души моей излучины  Пронзило терпкое вино.  > Пронзило терпкое winnо.  >  И перья страуса склоненные  > И перья страуса склоненные  В моем качаются мозгу,  > В моем качаются мозгу,  И очи синие бездонные  > И очи синие бездонные  Цветут на дальнем берегу.  > Цветут на дальнем берегу.  >  В моей душе лежит сокровище,  > В моей душе лежит сокровище,  И ключ поручен только мне!  > И ключ поручен только мне!  Ты право, пьяное чудовище!  > Ты право, пьяное чудовище!  Я знаю: истина в вине.  > Я знаю: истина в winnе. |

**Задание 7:** Прочитать текст Java-программы и все слова public в объявлении атрибутов и методов класса заменить на слово private.

Код решения приведен в листинге 13.

Листинг 13 — реализация решения

|  |
| --- |
| package org.example;  import java.io.\*;  import java.util.\*;  public class Main {  public static void main(String[] args) throws IOException {  System.out.println("=== File class work example ===\n\n");  System.out.println("Enter input file name (example.java): ");  Scanner scanner = new Scanner(System.in);  String in\_filename = scanner.nextLine();  if(Objects.equals(in\_filename, "")) {  in\_filename = "example.java";  System.out.println("using example.java");  }  System.out.println("Enter output file name (out\_example.java): ");  String out\_filename = scanner.nextLine();  if(Objects.equals(out\_filename, "")) {  out\_filename = "out\_example.java";  System.out.println("using out\_example.java");  }  out\_filename = "./output/" + out\_filename;  File out\_file = new File(out\_filename);  File dir = new File("./output");  dir.mkdirs();  out\_file.createNewFile();  BufferedReader reader = new BufferedReader(new FileReader(in\_filename));  BufferedWriter writer = new BufferedWriter(new FileWriter(out\_filename));  String currentLine;  System.out.println("File compare:");  while ((currentLine = reader.readLine()) != null) {  System.out.printf("%s\n", currentLine);  String new\_str = currentLine.replace("public ", "private ").replace("public\t", "private\t");  System.out.printf("> %s\n", new\_str);  writer.write(new\_str + System.lineSeparator());  }  writer.close();  reader.close();  }  } |

Результат исполнения программы приведен в листинге 14.

Листинг 14 — результат исполнения программы

|  |
| --- |
| === File class work example ===  Enter input file name (example.java):  using example.java  Enter output file name (out\_example.java):  using out\_example.java  File compare:  import java.io.\*;  > import java.io.\*;  import java.util.\*;  > import java.util.\*;  >  class Main {  > class Main {  >  public int i;  > private int i;  >  private String s;  > private String s;  >  public static void main(String[] args) throws IOException {  > private static void main(String[] args) throws IOException {  >  }  > }  }  > }  >  class MyClass {  > class MyClass {  public void doNothing(){  > private void doNothing(){  >  }  > }  >  private long m;  > private long m;  >  public boolean fl;  > private boolean fl;  }  > } |

**Задание 8:** Прочитать текст Java-программы и записать в другой файл в обратном порядке символы каждой строки.

Код решения приведен в листинге 15.

Листинг 15 — реализация решения

|  |
| --- |
| package org.example;  import java.io.\*;  import java.util.\*;  public class Main {  public static void main(String[] args) throws IOException {  System.out.println("=== File class work example ===\n\n");  System.out.println("Enter input file name (example.java): ");  Scanner scanner = new Scanner(System.in);  String in\_filename = scanner.nextLine();  if(Objects.equals(in\_filename, "")) {  in\_filename = "example.java";  System.out.println("using example.java");  }  System.out.println("Enter output file name (out\_example.java): ");  String out\_filename = scanner.nextLine();  if(Objects.equals(out\_filename, "")) {  out\_filename = "out\_example.java";  System.out.println("using out\_example.java");  }  out\_filename = "./output/" + out\_filename;  File out\_file = new File(out\_filename);  File dir = new File("./output");  dir.mkdirs();  out\_file.createNewFile();  BufferedReader reader = new BufferedReader(new FileReader(in\_filename));  BufferedWriter writer = new BufferedWriter(new FileWriter(out\_filename));  String currentLine;  System.out.println("File compare:");  while ((currentLine = reader.readLine()) != null) {  System.out.printf("%s\n", currentLine);  String new\_str = new StringBuilder(currentLine).reverse().toString();  System.out.printf("> %s\n", new\_str);  writer.write(new\_str + System.lineSeparator());  }  writer.close();  reader.close();  }  } |

Результат исполнения программы приведен в листинге 16.

Листинг 16 — результат исполнения программы

|  |
| --- |
| === File class work example ===  Enter input file name (example.java):  using example.java  Enter output file name (out\_example.java):  using out\_example.java  File compare:  import java.io.\*;  > ;\*.oi.avaj tropmi  import java.util.\*;  > ;\*.litu.avaj tropmi  >  class Main {  > { niaM ssalc  >  public int i;  > ;i tni cilbup  >  private String s;  > ;s gnirtS etavirp  >  public static void main(String[] args) throws IOException {  > { noitpecxEOI sworht )sgra ][gnirtS(niam diov citats cilbup  >  }  > }  }  > }  >  class MyClass {  > { ssalCyM ssalc  public void doNothing(){  > {)(gnihtoNod diov cilbup  >  }  > }  >  private long m;  > ;m gnol etavirp  >  public boolean fl;  > ;lf naeloob cilbup  }  > } |

**Вывод:** в ходе лабораторной работы были освоены базовые принципы работы с исключениями и файлами на языке Java.