|  |  |
| --- | --- |
|  | **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 **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**по лабораторной работе № 5**

**Вариант № 4**

**Название:** Исключения, файлы

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | С.А. Гисич |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цель:**

**Задание :** 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.

Листинг 1 – Обработка пациентов с исключениями

import java.lang.reflect.Array;

import java.util.ArrayList;

import java.util.InputMismatchException;

import java.util.Random;

import java.util.Scanner;

public class laba5\_3 {

public static void main(String[] args) {

System.out.println("Введите количество пациентов: ");

Scanner sc = new Scanner(System.in);

try {

int n = sc.nextInt();

Patient[] patients = new Patient[n];

Random random = new Random();

Diagnosis diagnosis;

for (int i = 0; i < n; i++) {

switch (random.nextInt(5)) {

case (0):

diagnosis = Diagnosis.ANOREXIA;

break;

case (1):

diagnosis = Diagnosis.AUTISM;

break;

case (2):

diagnosis = Diagnosis.BRONCHITIS;

break;

case (3):

diagnosis = Diagnosis.MIGRAINE;

break;

default:

diagnosis = null; // diagnosis = Diagnosis.HEADACHE;

}

patients[i] = new Patient((char) (random.nextInt(20) + 97) + "", (char) (random.nextInt(20) + 97) + "", (char) (random.nextInt(20) + 97) + "",(char) (random.nextInt(20) + 97) + "",(random.nextInt(1000)), (random.nextInt(1000)), diagnosis);

System.out.println(patients[i].toString());

}

try {

diagnosePatients(patients, Diagnosis.BRONCHITIS);

diagnosePatients(patients, Diagnosis.AUTISM);

diagnosePatients(patients, Diagnosis.ANOREXIA);

diagnosePatients(patients, Diagnosis.MIGRAINE);

searchPatients(patients, 200, 500);

} catch (NullPointerException e) {

System.out.println("Не может быть пациента без диагноза!");

}

} catch (InputMismatchException e){

System.out.println("Попробуйте указать количество пациентов с помощью цифр!");

}

}

public static ArrayList<Patient> diagnosePatients(Patient[] patients, Diagnosis diagnosis) {

System.out.println("--------------------------");

System.out.println("Пациенты с диагнозом - " + diagnosis.name());

ArrayList<Patient> diagnosedPatients = new ArrayList<>();

for (Patient patient: patients) {

if (patient.diagnosis.equals(diagnosis)) {

diagnosedPatients.add(patient);

System.out.println(patient.toString());

}

}

return diagnosedPatients;

}

public static ArrayList<Patient> searchPatients(Patient[] patients, int start, int end) {

System.out.println("--------------------------");

if (start > end) {

System.out.println("Не корректный диапазон");

return null;

}

System.out.println("Пациенты с номером медецинской карты в диапаззоне от " + start + " до " + end);

ArrayList<Patient> searchedPatients = new ArrayList<>();

for (Patient patient: patients) {

if (patient.medicalCardNumber > start && patient.medicalCardNumber < end) {

searchedPatients.add(patient);

System.out.println(patient.toString());

}

}

return searchedPatients;

}

}

**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки обработки исключений и файлов в java