Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования   
«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Отчет

По лабораторной работе №3

**«Первая программа на Java»**

по дисциплине «Программирование – 3 семестр»

Студент группы ПИ-12 Тришин. Я. А.

Преподаватель доцент, к.т.н. Потупчик А. И.

Барнаул 2022

Задание: написать программу для учёта больных, болезней, врачей и поликлиник на Java.

Программа:

import java.io.\*;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.nio.file.StandardOpenOption;

import java.util.Scanner;

import java.io.FileOutputStream;

import java.io.InputStreamReader;

//Пациент

class patient

{

private int id=0;

private String name = "";

private String diagnosis ="";

private String state = "";

public patient(int new\_id, String new\_patient\_name, String new\_diagnosis, String new\_state)

{

id = new\_id;

name= new\_patient\_name;

diagnosis=new\_diagnosis;

state=new\_state;

}

public patient()

{

int id = 0;

final String name = "";

final String diagnosis = "";

final String state = "";

}

public final void close()

{

}

//Вывод строки

public final void out(patient[] patients)

{

int i = 0;

System.out.printf("|id |ФИО пациета |Диагноз пациента|Состояние пациента|\n");

while (patients[i]!=null&&patients[i].id > 0)

{

System.out.printf("|%3d|%24s|%16s|%18s|\n", patients[i].id, patients[i].name, patients[i].diagnosis, patients[i].state);

i++;

}

System.out.printf("------------------------------------------------------------------\n");

}

//Ввод строки

public final void input\_patient(int total\_lines, patient[] patients, String patientname, String illnessname, String statusname)

{

patients[total\_lines - 1].id = total\_lines;

patients[total\_lines - 1].name= patientname;

patients[total\_lines - 1].diagnosis= illnessname;

patients[total\_lines - 1].state= statusname;

}

//Вывод результатов в файл//

public final void write\_results(patient[] patients)

{

Path filePath=Paths.get("", "", "patients.txt");

int i = 0;

try{

FileWriter writer = new FileWriter("patients.txt", true);

Files.writeString(filePath,String.format("|id |ФИО пациета |Диагноз пациента|Состояние пациента|\n"),StandardOpenOption.APPEND);

while (patients[i].id > 0)

{

Files.writeString(filePath,String.format("|%3d|%24s|%16s|%18s|\n", patients[i].id, patients[i].name, patients[i].diagnosis, patients[i].state),StandardOpenOption.APPEND);

i++;

}

Files.writeString(filePath,String.format("------------------------------------------------------------------\n"),StandardOpenOption.APPEND);

}

catch (IOException e) {

e.printStackTrace();

}

}

//Удаление строки

public final void patient\_del(int line, patient[] patients)

{

int i = 0, checks= 0;

while (patients[i].id > 0)

{

if (patients[i].id == line)

{

checks= 1;

}

if (checks==1)

{

patients[i].name=patients[i+1].name;

patients[i].diagnosis=patients[i+1].diagnosis;

patients[i].state=patients[i+1].state;

if(patients[i].id!=0)

patients[i].id = i + 1;

}

i++;

}

patients[i-1].id=0;

}

}

//Болезнь

class sickness

{

private int id=0;

private String diagnosis = "";

private String explanation = "";

public sickness(int new\_id, String new\_diagnosis, String new\_explanation)

{

id = new\_id;

diagnosis= new\_diagnosis;

explanation= new\_explanation;

}

public sickness()

{

int id = 0;

final String diagnosis = "";

final String explanation = "";

}

public final void close()

{

}

//Удаление строки

public final void ill\_del(int line, sickness[] ill)

{

int i = 0, checks= 0;

while (ill[i].id > 0)

{

if (ill[i].id == line)

{

checks= 1;

}

if (checks==1)

{

ill[i].diagnosis=ill[i+1].diagnosis;

ill[i].explanation=ill[i+1].explanation;

if(ill[i].id!=0)

ill[i].id = i + 1;

}

i++;

}

ill[i-1].id=0;

}

//Ввод строки

public final void ill\_in(int total\_lines, sickness[] ill, String illnessname) throws IOException

{

Scanner in = new Scanner(System.in,"ibm-866");

int check = 0;

int ill\_end = 0;

int i = 0;

int ch=ill[i].diagnosis.length();

while (ill[i]!=null&&ill[i].id > 0)

{

if (illnessname.compareToIgnoreCase (ill[i].diagnosis) == 0)

{

check = 1;

}

i++;

}

if(ill[total\_lines - 1]!=null)

{

ill[total\_lines - 1].diagnosis= illnessname;

ill[total\_lines - 1].id=total\_lines;

}

if (check == 0)

{

System.out.printf("Был введен новый диагноз, введите его описание:\n");

do

{

ill[total\_lines - 1].explanation=in.nextLine();

ch=ill[total\_lines - 1].explanation.length();

} while (ch < 1);

}

}

//Вывод

public final void out\_illness(int line, sickness[] ill)

{

for (int i = 0; i < 30; i++)

{

if (ill[i]!=null&&ill[i].id == line)

{

System.out.printf("%s - %s\n", ill[i].diagnosis, ill[i].explanation);

}

}

}

}

//Доктор

class doctor

{

private int id=0;

private String patient\_name = "";

private String doctor\_name = "";

public doctor()

{

int id = 0;

final String patient\_name = "";

final String doctor\_name = "";

}

public doctor(int new\_id, String new\_patient\_name, String new\_doctor\_name)

{

id = new\_id;

patient\_name= new\_patient\_name;

doctor\_name= new\_doctor\_name;

}

//Удаление строки

public final void doc\_del(int line, doctor[] doc)

{

int i = 0;

int check = 0;

while (doc[i+1]!=null&&doc[i]!=null&&doc[i].id > 0)

{

if (doc[i].id == line)

{

check = 1;

}

if (check != 0)

{

doc[i]=doc[i + 1];

if(doc[i].id!=0)

doc[i].id =i+1;

}

i++;

}

doc[i -1].id=0;

}

public final void close()

{

}

//Вывод в файл

public final void write\_doc(int i, doctor[] doc)

{

Path filePath=Paths.get("", "", "staff.txt");

try{

FileWriter writer = new FileWriter("staff.txt", true);

Files.writeString(filePath,String.format("|%24s ", doc[i].patient\_name),StandardOpenOption.APPEND);

}

catch (IOException e) {

e.printStackTrace();

}

}

public final void input\_doc(int total\_lines, doctor[] doc, String doctorname, String patientname)

{

doc[total\_lines - 1].id = total\_lines - 1;

doc[total\_lines - 1].doctor\_name=doctorname;

doc[total\_lines - 1].patient\_name= patientname;

}

////Вывод данных о докторе//

public final void out\_doc(int line, doctor[] doc)

{

System.out.printf("|Пациент |Доктор |\n");

System.out.printf("|%24s |%24s|\n",doc[line-1].patient\_name,doc[line - 1].doctor\_name);

System.out.printf("----------------------------------------------------\n");

}

}

//Больница

class hospital

{

private int id=0;

private String doctor\_name ="";

private String hospital\_name = "";

public hospital()

{

int id = 0;

final String doctor\_name = "";

final String hospital\_name = "";

}

public final void get\_doctor\_name(String new\_doc\_name)

{

doctor\_name= new\_doc\_name;

}

public hospital(int new\_id, String new\_doctor\_name, String new\_hospital\_name)

{

id = new\_id;

doctor\_name= new\_doctor\_name;

hospital\_name= new\_hospital\_name;

}

//Ввод

public final void input\_hosp(int total\_lines, hospital[] hosp, String doctorname, String hospitalname)

{

hosp[total\_lines - 1].id = total\_lines - 1;

hosp[total\_lines - 1].doctor\_name= doctorname;

hosp[total\_lines - 1].hospital\_name= hospitalname;

}

//вывод в файл

public final void write\_hosp(int i, hospital[] hosp)

{

Path filePath=Paths.get("", "", "staff.txt");

try{

FileWriter writer = new FileWriter("staff.txt", true);

Files.writeString(filePath,String.format("|%24s|%24s |\n", hosp[i].doctor\_name,hosp[i].hospital\_name),StandardOpenOption.APPEND);

}

catch (IOException e) {

e.printStackTrace();

}

}

public final void close()

{

}

//Удаление строки

public final void hosp\_del(int line, hospital[] hosp)

{

int i = 0;

int check = 0;

while (hosp[i+1]!=null&&hosp[i]!=null&&hosp[i].id > 0)

{

if (hosp[i].id == line)

{

check = 1;

}

if (check != 0)

{

hosp[i]=hosp[i + 1];

if(hosp[i].id!=0)

hosp[i].id =i+1;

}

i++;

}

hosp[i -1].id=0;

}

//Вывод данных о больнице//

public final void out\_hospital(int line, hospital[] hosp)

{

System.out.printf("|Доктор |Больница |\n");

System.out.printf("|%24s|%24s|\n", hosp[line - 1].doctor\_name, hosp[line - 1].hospital\_name);

System.out.printf("----------------------------------------------------\n");

}

}

//Состояние пациента

class status

{

private int id=0;

private String patient\_status="";

private String status\_description="";

public status()

{

int id=0;

String patient\_status = "";

String status\_description = "";

}

public status(int new\_id, String new\_patient\_status, String new\_status\_description)

{

id = new\_id;

patient\_status= new\_patient\_status;

status\_description= new\_status\_description;

}

//Вывод

public final void out\_state(int line, status[] state)

{

System.out.printf("%s - %s \n", state[line-1].patient\_status, state[line-1].status\_description);

}

//Ввод

public final void new\_line(int total\_lines, String statusname, status[] state)

{

int i;

i=4;

if (statusname.compareToIgnoreCase("Здоров") == 0)

{

i = 0;

}

if (statusname.compareToIgnoreCase("Умеренное") == 0)

{

i = 1;

}

if (statusname.compareToIgnoreCase("Серьёзное") == 0)

{

i = 2;

}

switch (i)

{

case 0:

state[total\_lines - 1] = state[30];

state[total\_lines - 1].id = total\_lines - 1;

break;

case 1:

state[total\_lines - 1] = state[31];

state[total\_lines - 1].id = total\_lines - 1;

break;

case 2:

state[total\_lines - 1] = state[32];

state[total\_lines - 1].id = total\_lines - 1;

break;

default:

break;

}

}

//Удаление строки

public final void del\_line(int line, status[] state)

{

int i = 0;

int check = 0;

while (state[i].id > 0 && state[i].id < 30)

{

if (state[i].id == line)

{

check = 1;

}

if (check != 0)

{

state[i]=state[i + 1];

if(state[i].id!=0)

state[i].id =i+1;

}

i++;

}

state[i-1].id=0;

}

public final void close()

{

}

}

class Main

{

public static void main(String[] args) throws IOException

{

Scanner in = new Scanner(System.in,"ibm-866");

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

int line\_num,ch;

int total\_lines = 1;

int check\_lines = 0;

String patientname = new String(new char[40]);

String doctorname = new String(new char[40]);

String hospitalname = new String(new char[40]);

String statusname = new String(new char[20]);

String illnessname = new String(new char[40]);

patient[] patients=new patient[30];

patients[0] = new patient(1, "Алексей А.A.", "Грипп", "Здоров");

status[] state = new status[33];

for(int k=0;k<33;k++)

{

switch(k)

{

case 0:

state[k] = new status(1, "Здоров", "Пациент здоров");

break;

case 30:

state[k] = new status(30, "Здоров", "Пациент здоров");

break;

case 31:

state[k] = new status(31, "Умеренное", "Не наблюдается тяжёлых ослажнений");

break;

case 32:

state[k] = new status(32, "Серьёзное", "Необходимо постоянное наблюдение и уход за пациентом");

break;

default:

state[k] = new status();

break;

}

}

hospital[] hosp = new hospital[30];

hosp[0] = new hospital(1,"Борисов Ф.А.", "Городская поликлиника №12");

doctor[] doc = new doctor[30];

doc[0] = new doctor(1,"Александр И.И.","Борисов Ф.А.");

sickness[] ill = new sickness[30];

ill[0] = new sickness(1,"Грипп","Острое респираторное вирусное заболевание, вызываемое вирусами гриппа,поражающее верхние дыхательные пути");

for(int k=1;k<30;k++)

patients[k] = new patient();

for(int k=1;k<30;k++)

hosp[k] = new hospital();

for(int k=1;k<30;k++)

doc[k] = new doctor();

for(int k=1;k<30;k++)

ill[k] = new sickness();

int repeat = 1;

do

{

check\_lines = 0;

patients[0].out(patients);

System.out.print("Введите:\n1-для просмотра данных о пациенте\n2-для просмотра данных о болезни\n3-для просмотра состояния пациета\n4-для удаления строки\n5-для добавления строки\n6-Записать таблицу в файл\n7-закрыть программу\n");

int choice\_patient;

int choice\_doctor;

do

{

choice\_patient=in.nextInt();

} while (choice\_patient > 7 || choice\_patient < 1);

switch (choice\_patient)

{

//просмотр данных о пациенте

case 1:

if (total\_lines > 0)

{

System.out.print("Введите номер пациента:\n");

do

{

if (check\_lines != 0)

{

System.out.print("Повторите ввод\n");

}

line\_num=in.nextInt();

check\_lines = 1;

} while (line\_num > total\_lines || line\_num < 1);

doc[0].out\_doc(line\_num, doc);

System.out.print("Введите:\n1-Просмотр информации о докторе\nДля возвращение к предыдущей таблице введите любое другое число\n");

choice\_doctor=in.nextInt();

switch (choice\_doctor)

{

case 1:

hosp[0].out\_hospital(line\_num, hosp);

break;

default:

break;

}

}

else

{

System.out.print("Нет записей\n");

}

break;

//просмотр данных о болезни

case 2:

if (total\_lines > 0)

{

check\_lines = 0;

System.out.print("Введите номер пациента:\n");

do

{

if (check\_lines != 0)

{

System.out.print("Повторите ввод\n");

}

line\_num=in.nextInt();

check\_lines = 1;

} while (line\_num > total\_lines || line\_num < 1);

ill[0].out\_illness(line\_num, ill);

}

else

{

System.out.print("Нет записей\n");

}

break;

//просмотр состояния пациета

case 3:

check\_lines = 0;

if (total\_lines > 0)

{

System.out.print("Введите номер пациента:\n");

do

{

if (check\_lines != 0)

{

System.out.print("Повторите ввод\n");

}

line\_num=in.nextInt();

check\_lines = 1;

} while (line\_num > total\_lines || line\_num < 1);

state[0].out\_state(line\_num, state);

}

else

{

System.out.print("Нет записей\n");

}

break;

//удаление строки

case 4:

if (total\_lines > 0)

{

check\_lines = 0;

System.out.print("Введите номер строки для удаления:\n");

do

{

if (check\_lines != 0)

{

System.out.print("Повторите ввод\n");

}

line\_num=in.nextInt();

check\_lines = 1;

} while (line\_num > total\_lines || line\_num < 1);

patients[0].patient\_del(line\_num, patients);

hosp[0].hosp\_del(line\_num, hosp);

ill[0].ill\_del(line\_num,ill);

doc[0].doc\_del(line\_num, doc);

state[0].del\_line(line\_num, state);

total\_lines--;

}

else

{

System.out.print("Нет записей для удаления\n");

}

break;

//добавления строки

case 5:

total\_lines+=1;

in.nextLine();

System.out.print("Введите ФИО пациента:\n");

do

{

patientname=in.nextLine();

ch=doctorname.length();

} while (ch == 0);

System.out.print("Введите ФИО лечащего доктора пациента:\n");

do

{

doctorname=in.nextLine();

ch=doctorname.length();

} while (ch == 0);

doc[0].input\_doc(total\_lines,doc,doctorname,patientname);

System.out.print("Введите название больницы доктора:\n");

do

{

hospitalname=in.nextLine();

ch=hospitalname.length();

} while (ch == 0);

hosp[0].input\_hosp(total\_lines, hosp, doctorname, hospitalname);

System.out.print("Введите диагноз пациента:\n");

do

{

illnessname=in.nextLine();

ch=illnessname.length();

} while (ch == 0);

ill[0].ill\_in(total\_lines, ill, illnessname);

System.out.print("Введите состояние пациента (Здоров, Умеренное, Серьёзное):\n");

do

{

statusname=in.nextLine();

} while (statusname.compareToIgnoreCase("Здоров") != 0 && statusname.compareToIgnoreCase("Умеренное") !=0 && statusname.compareToIgnoreCase("Серьёзное") != 0);

state[0].new\_line(total\_lines,statusname,state);

patients[0].input\_patient(total\_lines, patients, patientname, illnessname, statusname);

break;

//Записать таблицу в файл

case 6:

File r=new File("staff.txt");

File w=new File("patients.txt");

r.delete();

w.delete();

w.createNewFile();

r.createNewFile();

FileWriter pat = new FileWriter(r, true);

FileWriter staff = new FileWriter(w, true);

Path filePath=Paths.get("", "", "staff.txt");

patients[0].write\_results(patients);

try{

Files.writeString(filePath,String.format("|Пациент |Доктор |Больница |\n"),StandardOpenOption.APPEND);

for (int i = 0; i < total\_lines; i++)

{

doc[0].write\_doc(i, doc);

hosp[0].write\_hosp(i, hosp);

}

Files.writeString(filePath,String.format("-------------------------------------------------------------------------------\n"),StandardOpenOption.APPEND);

}

catch (IOException e)

{

e.printStackTrace();

}

break;

//закрыть программу

case 7:

repeat = 0;

break;

default:

break;

}

} while (repeat != 0);

in.close();

patients = null;

ill = null;

state=null;

}

}

Проверка:





