МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

“ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Інститут ІКНІ

Кафедра автоматизовані системи управління



ЗВІТ

До лабораторної роботи №2

З курсу “Прикладне програмування”

Варіант: 12

**Виконав студент:**

Міщук М. В.

Групи КН-203

**Прийняв викладач:**

Вергун В. Р.

**Львів – 2020**

**Тема роботи**: Робота з класами

**Індивідуальне завдання:**

|  |  |
| --- | --- |
| **2** | **Student:** id, Прізвище, Ім'я, По батькові, Дата народження, Адреса, Телефон, Факультет, Курс, Група.  Скласти масив об'єктів. Вивести:   1. список студентів заданого факультету; 2. список студентів, які народились після заданого року; 3. список навчальної групи. |

**Хід роботи:**

1. Створив проект, що складається з двох класів: Main та Student. Кожний клас є розміщений у окремому пакеті. У класі Student визначив приватні поля для зберігання указаних даних, конструктори для створення об’єктів та відкриті методи setValue(), getValue(), toString() для доступу до полів об’єкту.
2. В основному класі програми визначити методи, що створюють масив об'єктів. Задати критерії вибору даних та вивести ці дані на консоль. Для кожного критерію створити окремий функціональний об’єкт.
3. Виконати програму, та пересвідчитись, що дані зберігаються та корректно виводяться на екран відповідно до вказаних критеріїв.

**Текст програми:**

**Main.java:**

package com.company.Main;

import com.company.Student.Student;

import java.util.Arrays;

import java.util.Calendar;

import java.util.GregorianCalendar;

import java.util.function.Consumer;

import java.util.function.Predicate;

public class Main {

public static void main(String[] args) {

var Faculty = "Photography art";

var Group = "SS42";

var YearAfter = 2001;

Consumer<Student[]> Print = (Student[] arr) -> {

for(Student a : arr){

if(a == null)

break;

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

}

System.out.println();

};

Student[] stud = CreateStudentArr();

Predicate<Student> BornAfter = x-> x.getBirthday().get(Calendar.YEAR) >= YearAfter;

Predicate<Student> OnFaculty = x-> x.getFaculty() == Faculty;

Predicate<Student> InGroup = x-> x.getGroup() == Group;

System.out.println();

System.out.println("Students born after " + String.valueOf(YearAfter) + ":");

Print.accept(Select(stud, BornAfter));

System.out.println("Students of " + Faculty + " faculty:");

Print.accept(Select(stud, OnFaculty));

System.out.println("Students of " + Group + " group:");

Print.accept(Select(stud, InGroup));

return;

}

private static Student[] CreateStudentArr() {

Student[] res = {

new Student(12345, 1, "Max", "","Caulfield", "Arcadia bay", "+011683934", "Photography art", "PA11", new GregorianCalendar(2002, 3, 25) ),

new Student(33939, 1, "Chloe", "Mae", "Price", "Arcadia bay", "+034857294", "Photography art", "PA11", new GregorianCalendar(2001, 4, 19) ),

new Student(84829, 3, "Rachel", "Renee", "Amber", "Arcadia bay", "+093867239", "Photography art", "PA31", new GregorianCalendar(2001, 10, 1) ),

new Student(39484, 3, "Nathan", "Joseph", "Prescott", "Arcadia bay", "+034395957", "Photography art", "PA31", new GregorianCalendar(2001, 7, 4) ),

new Student(23493, 2, "Kyle", "Lee", "Broflovski", "South Park", "+023485733", "Social studies", "SS24", new GregorianCalendar(2007, 2, 10) ),

new Student(34349, 2, "David", "Robert", "Madsen", "Arcadia bay", "+086729596", "Social studies", "SS24", new GregorianCalendar(1998, 1, 12) ),

new Student(23943, 4, "Stan", "James", "Marsh", "South Park", "+027439439", "Social studies", "SS42", new GregorianCalendar(2005, 2, 6) ),

new Student(98385, 4, "Randy", "", "Marsh", "South Park", "+086330712", "Social studies", "SS42", new GregorianCalendar(2000, 7, 20) ),

new Student(34803, 2, "Eric", "Joseph ", "Cartman", "South Park", "+025967284", "Philosophy", "PH24", new GregorianCalendar(2007, 9, 15) ),

new Student(39485, 2, "Victoria", "Lynn", "Chase", "Arcadia bay", "+037885375", "Philosophy", "PH24", new GregorianCalendar(2002, 5, 4) ),

new Student(96567, 3, "Kenny", "", "McCormick", "South Park", "+078340873", "Philosophy", "PH32", new GregorianCalendar(2004, 8, 14) ),

new Student(83757, 3, "Mark", "", "Jefferson", "Arcadia bay", "+096734967", "Philosophy", "PH34", new GregorianCalendar(1995, 8, 22) )

};

return res;

}

private static Student[] Select(Student[] stud, Predicate<Student> Pred){

int sz = stud.length, j = 0;

Student[] res = new Student[sz];

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

if(Pred.test(stud[i])){

res[j++] = stud[i];

}

}

return res;

}

}

**Student.java:**

package com.company.Student;

import java.util.Calendar;

import java.util.GregorianCalendar;

public class Student {

private int id;

private int Course;

private String FirstName;

private String MiddleName;

private String LastName;

private String Address;

private String Phone;

private String Faculty;

private String Group;

private GregorianCalendar Birthday;

public Student() {

this(0, 0, "Unknown", "", "Unknown",

"Unknown", "Unknown", "None", "None", new GregorianCalendar(1970 , 1, 1));

}

public Student(int id, int course, String firstName, String middleName, String lastName, String address, String phone, String faculty, String group, GregorianCalendar birthday) {

this.id = id;

Course = course;

FirstName = firstName;

MiddleName = middleName;

LastName = lastName;

Address = address;

Phone = phone;

Faculty = faculty;

Group = group;

Birthday = birthday;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getCourse() {

return Course;

}

public void setCourse(int course) {

Course = course;

}

public String getFirstName() {

return FirstName;

}

public void setFirstName(String firstName) {

FirstName = firstName;

}

public String getMiddleName() {

return MiddleName;

}

public void setMiddleName(String middleName) {

MiddleName = middleName;

}

public String getLastName() {

return LastName;

}

public void setLastName(String lastName) {

LastName = lastName;

}

public String getAddress() {

return Address;

}

public void setAddress(String address) {

Address = address;

}

public String getPhone() {

return Phone;

}

public void setPhone(String phone) {

Phone = phone;

}

public String getFaculty() {

return Faculty;

}

public void setFaculty(String faculty) {

Faculty = faculty;

}

public String getGroup() {

return Group;

}

public void setGroup(String group) {

Group = group;

}

public GregorianCalendar getBirthday() {

return Birthday;

}

public void setBirthday(GregorianCalendar birthday) {

Birthday = birthday;

}

@Override

public String toString() {

return FirstName + " " + MiddleName + " " + LastName + " (ID" + String.valueOf(id) + ")" + ", " +

String.valueOf(Birthday.get(Calendar.DAY\_OF\_MONTH))+"."+String.valueOf(Birthday.get(Calendar.MONTH))+"."+String.valueOf(Birthday.get(Calendar.YEAR))+"\n" +

Faculty + ", " + Group + ", " + String.valueOf(Course) + " course" + "\n" +

Address + "\n" +

Phone + "\n";

}

}

**Результат виконання програми:**

Students born after 2001:

**Max Caulfield** (ID12345), 25.3.2002

Photography art, PA11, 1 course

Arcadia bay

+011683934

**Chloe Mae Price** (ID33939), 19.4.2001

Photography art, PA11, 1 course

Arcadia bay

+034857294

**Rachel Renee** Amber (ID84829), 1.10.2001

Photography art, PA31, 3 course

Arcadia bay

+093867239

**Nathan Joseph Prescott** (ID39484), 4.7.2001

Photography art, PA31, 3 course

Arcadia bay

+034395957

**Kyle Lee Broflovski** (ID23493), 10.2.2007

Social studies, SS24, 2 course

South Park

+023485733

**Stan James Marsh** (ID23943), 6.2.2005

Social studies, SS42, 4 course

South Park

+027439439

**Eric Joseph Cartman** (ID34803), 15.9.2007

Philosophy, PH24, 2 course

South Park

+025967284

**Victoria Lynn Chase** (ID39485), 4.5.2002

Philosophy, PH24, 2 course

Arcadia bay

+037885375

**Kenny McCormick** (ID96567), 14.8.2004

Philosophy, PH32, 3 course

South Park

+078340873

Students of Photography art faculty:

**Max Caulfield** (ID12345), 25.3.2002

Photography art, PA11, 1 course

Arcadia bay

+011683934

**Chloe Mae Price** (ID33939), 19.4.2001

Photography art, PA11, 1 course

Arcadia bay

+034857294

**Rachel Renee Amber** (ID84829), 1.10.2001

Photography art, PA31, 3 course

Arcadia bay

+093867239

**Nathan Joseph Prescott** (ID39484), 4.7.2001

Photography art, PA31, 3 course

Arcadia bay

+034395957

Students of SS42 group:

**Stan James Marsh** (ID23943), 6.2.2005

Social studies, SS42, 4 course

South Park

+027439439

**Randy Marsh** (ID98385), 20.7.2000

Social studies, SS42, 4 course

South Park

+086330712

**Висновок**: виконуючи лабораторну я навчився працювати з класами Java.