**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине: **«**Разработка приложений для *IPhone* и *IPad***»**

на тему: **Основы программирования на языке Swift**

Выполнил: студент гр. ИТП-41

Солодков М.А

Приняла: Семенченя Т.С.

Гомель 2020

**Цель**: изучить основы языка программирование Swift.

**Ход работы**

**Задание**

Создать классы, спецификации которых приведены ниже.

Определить дополнительно методы в классе, создающем массив объектов. Задать критерий выбора данных и вывести эти данные на консоль.

**Вариант 13**

13. Student: id, Фамилия, Имя, Отчество, Дата Рождения, Адрес, Телефон.

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;

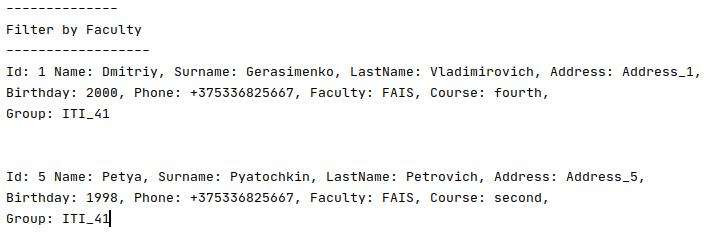
**Результат выполнения:**



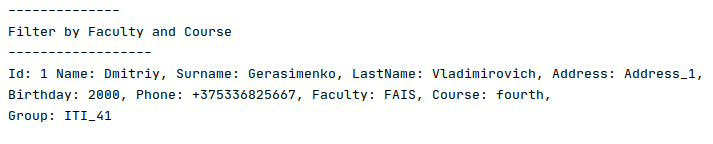
**Рисунок 1** – Данные о студентах



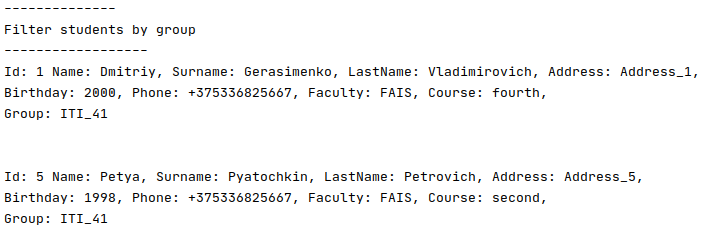
**Рисунок 2** – Вывод всех добавленных студентов



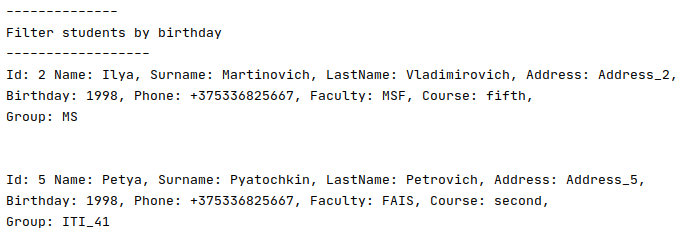
**Рисунок 3** – Фильтрация студентов по заданному факультету



**Рисунок 4** – Фильтрация студентов по заданному факультету и заданному курсу



**Рисунок 5 –** Фильтрация студентов по заданной группе



**Рисунок 6 –** Фильтрация студентов, у которых год рождения больше указанного

**Вывод:** в ходе выполнения лабораторной работы №2 были изучены основы программирования на языке *Swift*. А так же ознакомлен с работой с классами.

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

**Student.swift**

enum Faculty{

case MSF, MTF, FAIS, EF

}

enum Course{

case first, second, threed, fourth, fifth

}

enum Group{

case ITP\_41, ITI\_41, MS, MT, EF\_41, IP\_41

}

class Student{

var id: Int;

var name: String;

var surname: String;

var lastName: String;

var address: String;

var birthday: Int;

var phone: String;

var faculty: Faculty;

var course: Course;

var group: Group;

init(name: String, surname: String, lastName: String, address: String, birthday: Int,

phone: String, faculty: Faculty, course: Course, group: Group){

self.id = 0;

self.name = name;

self.surname = surname;

self.lastName = lastName;

self.address = address;

self.phone = phone;

self.faculty = faculty;

self.birthday = birthday;

self.course = course;

self.group = group;

}

func setId(id: Int){

self.id = id;

}

var description: String {

return """

Id: \(id) Name: \(name), Surname: \(surname), LastName: \(lastName), Address: \(address),

Birthday: \(birthday), Phone: \(phone), Faculty: \(faculty), Course: \(course),

Group: \(group)

"""

}

}

**University.swift**

enum StudentFilter{

case filterByFaculty(students: [Student], faculty: Faculty)

case filterByFacultyAndCourse(students: [Student], faculty: Faculty, course: Course)

case filterByBirthdayStudent(students: [Student], birthday: Int)

case filterByGroup(students: [Student], group: Group)

func filter() -> [Student]{

switch self{

case .filterByFaculty(let students, let faculty):

return students.filter({$0.faculty == faculty})

case .filterByFacultyAndCourse(let students, let faculty, let course):

return students.filter({$0.faculty == faculty && $0.course == course})

case .filterByBirthdayStudent(let students, let birthday):

return students.filter({$0.birthday < birthday})

case .filterByGroup(let students, let group):

return students.filter({$0.group == group})

}

}

}

class University{

var students: [Student];

static var currentStudent: Int = 0;

init(){

self.students = [Student]();

}

func addStudent(student: Student){

University.currentStudent += 1;

student.setId(id: University.currentStudent);

self.students.append(student);

}

func filterByFaculty(faculty: Faculty) -> [Student]{

return StudentFilter.filterByFaculty(students: self.students, faculty: faculty).filter();

}

func filterByFacultyAndCourse(faculty: Faculty, course: Course) -> [Student]{

return StudentFilter.filterByFacultyAndCourse(students: self.students, faculty: faculty, course: course).filter();

}

func filterByBirthdayStudent(birthday: Int) -> [Student]{

return StudentFilter.filterByBirthdayStudent(students: self.students, birthday: birthday).filter();

}

func filterByGroup(group: Group) -> [Student]{

return StudentFilter.filterByGroup(students: self.students, group: group).filter();

}

}

**Main.swift**

var student\_1 = Student(name: "Dmitriy", surname: "Gerasimenko", lastName: "Vladimirovich",

address: "Address\_1", birthday: 2000, phone: "+375336825667",

faculty: Faculty.FAIS, course: Course.fourth, group: Group.ITI\_41)

var student\_2 = Student(name: "Ilya", surname: "Martinovich", lastName: "Vladimirovich",

address: "Address\_2", birthday: 1998, phone: "+375336825667",

faculty: Faculty.MSF, course: Course.fifth, group: Group.MS)

var student\_3 = Student(name: "Anastasya", surname: "Hilchenko", lastName: "Alexandrovna",

address: "Address\_3", birthday: 1999, phone: "+375336825667",

faculty: Faculty.EF, course: Course.fifth, group: Group.MT)

var student\_4 = Student(name: "Ilya", surname: "Nazarenko", lastName: "Vladimirovich",

address: "Address\_4", birthday: 1999, phone: "+375336825667",

faculty: Faculty.MSF, course: Course.fourth, group: Group.MT)

var student\_5 = Student(name: "Petya", surname: "Pyatochkin", lastName: "Petrovich",

address: "Address\_5", birthday: 1998, phone: "+375336825667",

faculty: Faculty.FAIS, course: Course.second, group: Group.ITI\_41)

let university = University();

university.addStudent(student: student\_1)

university.addStudent(student: student\_2)

university.addStudent(student: student\_3)

university.addStudent(student: student\_4)

university.addStudent(student: student\_5)

print("All students: ")

print("--------------")

for student in university.students{

print(student.description);

print("\n");

}

print("--------------")

print("Filter by Faculty");

print("------------------")

var filterStudents = university.filterByFaculty(faculty: Faculty.FAIS)

for student in filterStudents{

print(student.description);

print("\n");

}

print("--------------")

print("Filter by Faculty and Course");

print("------------------")

filterStudents = university.filterByFacultyAndCourse(faculty: Faculty.FAIS, course: Course.fourth)

for student in filterStudents{

print(student.description);

print("\n");

}

print("--------------")

print("Filter students by group");

print("------------------")

filterStudents = university.filterByGroup(group: Group.ITI\_41)

for student in filterStudents{

print(student.description);

print("\n");

}

print("--------------")

print("Filter students by birthday");

print("------------------")

filterStudents = university.filterByBirthdayStudent(birthday: 1999)

for student in filterStudents{

print(student.description);

print("\n");

}