Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра инженерной психологии и эргономики

­­­­­

Современные языки программирования

Отчет по лабораторной работе №1

«Использование языка программирования Kotlin»

Выполнил: Поливкин В.С.

Студент группы 310902

Преподаватель: Усенко Ф.В.

Минск 2024

***Индивидуальное задания для лабораторной работы:***

Классу Университет добавить методы Провести лекцию(), Провести семинар(), Провести лабораторный практикум(). Эти методы должны менять поля знания и оценки класса Студент. Добавить класс Детский сад.

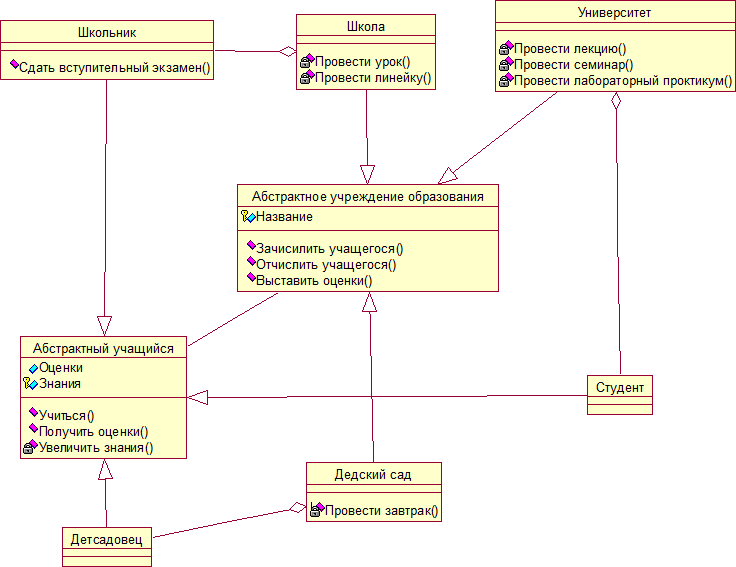


Рисунок 1 – Диаграмма классов

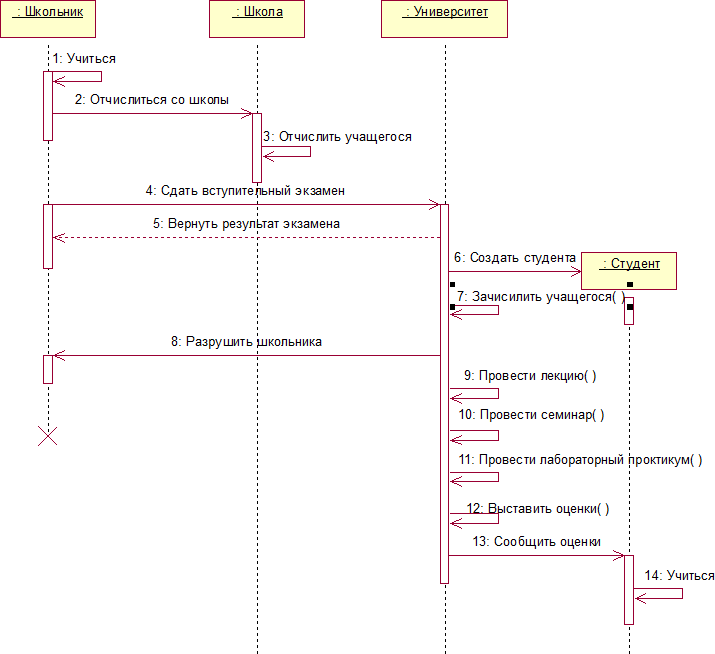


Рисунок 2 – Диаграмма последовательности

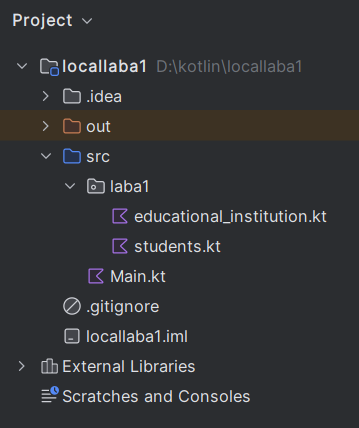


Рисунок 2 – Структура проекта.

Код программы:

Educational\_institution.kt:

package laba1.laba1

abstract open class Educational\_institutions(name:String="Undefined"){

internal var name:String=name

public fun enroll(pupil:Pupil,){//зачислить

println("${pupil.javaClass.simpleName} зачислен в $name.")

}

public fun expel(pupil:Pupil){//отчислить

println("${pupil.javaClass.simpleName} отчислен из $name.")

}

open public fun set\_ratings(pupil: Pupil) {

if(pupil.knowledge>20)

pupil.marks.add(10)

else if(pupil.knowledge==0&&pupil.knowledge==1)

pupil.marks.add(1)

else

pupil.marks.add(pupil.knowledge/2)

println("Оценки ${pupil.javaClass.simpleName} выставлены")

}

}

class University(name: String):Educational\_institutions(name){

override fun set\_ratings(pupil: Pupil) {

println("Где вы хотите выставить оценки?")

println("1.На лекции")

println("2.На семинаре")

println("3.На лабораторной")

var a:Boolean=true

while(a) {

var ch = readln().toInt()

when (ch) {

1 -> {give\_lecture(pupil)

a= false}

2 -> {

give\_seminar(pupil)

a=false

}

3 -> {give\_laba(pupil)

a= false

}

else -> println("невверные значения")

}

}

super.set\_ratings(pupil)

}

private fun give\_lecture(pupil:Pupil){

pupil.knowledge += 2

println("Лекция проведена в $name. Знания студента увеличены.")

}

private fun give\_seminar (pupil:Pupil){

pupil.knowledge += 1

println("Семинар проведен в $name. Знания студента увеличены.")

}

private fun give\_laba (pupil:Pupil){

pupil.knowledge += 2

println("Лабораторная проведена в $name. Знания студента увеличены.")

}

}

class School(name: String):Educational\_institutions(name){

override fun set\_ratings(pupil: Pupil) {

println("Где вы хотите выставить оценки?")

println("1.На уроке")

println("2.На линейке")

var a:Boolean=true

while(a) {

var ch = readln().toInt()

when (ch) {

1 -> {give\_lesson(pupil)

a=false}

2 -> {

give\_meeting(pupil)

a= false

}

else -> println("невверные значения")

}

}

super.set\_ratings(pupil)

}

private fun give\_lesson (pupil:Pupil){

pupil.knowledge += 2

println("Урок проведен в $name. Знания Школьника увеличены.")

}

private fun give\_meeting (pupil:Pupil){

println("Линейка проведена в $name")

}

}

class Kindergarten(name: String):Educational\_institutions(name){

override fun set\_ratings(pupil: Pupil) {

println("В саду нет оценок,можите провести завтрак")

println("1.Провести завтрак")

println("2.Нет")

var a:Boolean=true

while(a) {

var ch = readln().toInt()

when (ch) {

1 -> {give\_breakfast()

a= false}

2 -> {

a=false

}

else -> println("невверные значения")

}

}

}

private fun give\_breakfast(){

println("Завтрак проведен в $name.")

}

}

students.kt:

package laba1.laba1

abstract class Pupil(){

public var marks:MutableList<Int> = mutableListOf()

internal var knowledge:Int=0;

public fun study(){

println("${this.javaClass.simpleName} учится.")

improve\_knoledge();

}

public fun get\_mark():MutableList<Int>{

return this.marks

}

private fun improve\_knoledge(){

knowledge += 2

}

}

class Student:Pupil(){

}

class Schoolchild:Pupil(){

public fun take\_exam():Boolean{

if (knowledge > 0) {

println("Школьник сдал экзамен.")

return true

}

println("Школьник не сдал экзамен.")

return false

}

}

class Kindergartener:Pupil(){

}

Main.kt:

package laba1.laba1

import laba1.\*

fun main(args:Array<String>) {

// Создание одного экземпляра каждого типа ученика и учебного заведения

val student = Student()

val schoolchild = Schoolchild()

val kindergartener = Kindergartener()

val school = School("Школа")

val university = University("Университет")

val kindergarten = Kindergarten("Детский сад")

var flag = true

while (flag) {

println("Выберите действие:")

println("1. Студент")

println("2. Школьник")

println("3. Детсадовец")

println("4. Универ")

println("5. Школа")

println("6. Детский сад")

println("7. Выход")

val choice = readLine()!!.toInt()

when (choice) {

1 -> studentMenu(student)

2 -> schoolchildMenu(schoolchild,student,university,school)

3 -> kindergartenerMenu(kindergartener)

4 -> universityMenu(university,student)

5 -> schoolMenu(school,schoolchild)

6 -> kindergartenMenu(kindergarten,kindergartener)

7 -> flag = false

else -> println("Неверный выбор")

}

}

}

fun studentMenu(student: Student) {

println("Меню студента:")

println("1. Учиться")

println("2. Посмотреть оценки")

println("3. Выход")

val choice = readLine()!!.toInt()

when (choice) {

1 -> student.study()

2 -> viewMarks(student)

3 -> println("Выход из меню студента")

else -> println("Неверный выбор")

}

}

fun schoolchildMenu(schoolchild: Schoolchild,student: Student,university: University,scholl:School) {

println("Меню школьника:")

println("1. Учиться")

println("2. Сдать экзамен")

println("3. Посмотреть оценки")

println("4. Выход")

val choice = readLine()!!.toInt()

when (choice) {

1 -> schoolchild.study()

2 -> { if(schoolchild.take\_exam()){

scholl.expel(schoolchild);

university.expel(student);

student.marks=schoolchild.get\_mark();

student.knowledge=schoolchild.knowledge;

schoolchild.knowledge=0;

schoolchild.marks= mutableListOf();

}

}

3 -> viewMarks(schoolchild)

4 -> println("Выход из меню школьника")

else -> println("Неверный выбор")

}

}

fun kindergartenerMenu(kindergartener: Kindergartener) {

println("Меню детсадовца:")

println("1. Учиться")

println("2. Посмотреть оценки")

println("3. Выход")

val choice = readLine()!!.toInt()

when (choice) { 1 -> kindergartener.study()

2 -> viewMarks(kindergartener)

3 -> println("Выход из меню детсадовца")

else -> println("Неверный выбор")

}

}

fun universityMenu(university: University,student: Student) {

println("Меню университета:")

println("1. Зачислить студента")

println("2. Отчислить студента")

println("3. Выставить оценки студенту")

println("4. Посмотреть оценки студента")

println("5. Выход")

val choice = readLine()!!.toInt()

when (choice) {

1 -> university.enroll(student)

2 -> university.expel(student)

3 -> university.set\_ratings(student)

4 -> viewMarks(student)

5 -> println("Выход из меню университета")

else -> println("Неверный выбор")

}

}

fun schoolMenu(school: School,schoolchild: Schoolchild) {

println("Меню школы:")

println("1. Зачислить школьника")

println("2. Отчислить школьника")

println("3. Выставить оценки школьнику")

println("4. Посмотреть оценки школьника")

println("5. Выход")

val choice = readLine()!!.toInt()

when (choice) {

1 -> school.enroll(schoolchild)

2 -> school.expel(schoolchild)

3 -> school.set\_ratings(schoolchild)

4 -> viewMarks(schoolchild)

5 -> println("Выход из меню школы")

else -> println("Неверный выбор")

}

}

fun kindergartenMenu(kindergarten: Kindergarten,kindergartener: Kindergartener) {

println("Меню детского сада:")

println("1. Зачислить детсадовца")

println("2. Отчислить детсадовца")

println("3. Выставить оценки детсадовцу")

println("4. Выход")

val choice = readLine()!!.toInt()

when (choice) {

1 -> kindergarten.enroll(kindergartener)

2 -> kindergarten.expel(kindergartener)

3 -> kindergarten.set\_ratings(kindergartener)

4-> println("Выход из меню детского сада")

else -> println("Неверный выбор")

}

}

fun viewMarks( pupil: Pupil) {

println("Оценки ${pupil.javaClass.simpleName}: ${pupil.get\_mark()}")

}

Результат выполнения программы представлен на рисунках 3-6.

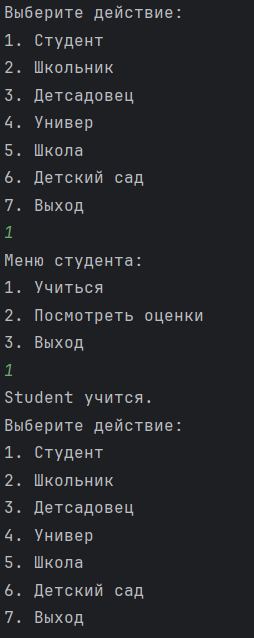


Рисунок 3 – Результат выполнения программы.

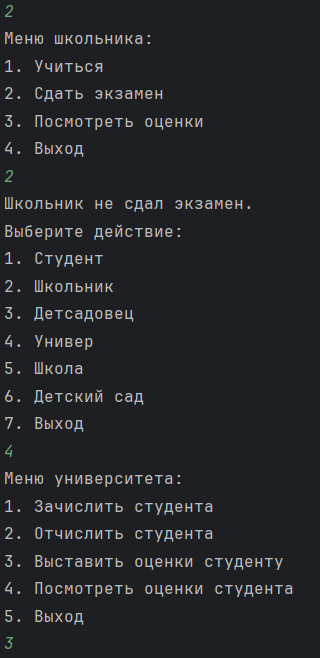


Рисунок 4 – Результат выполнения программы.

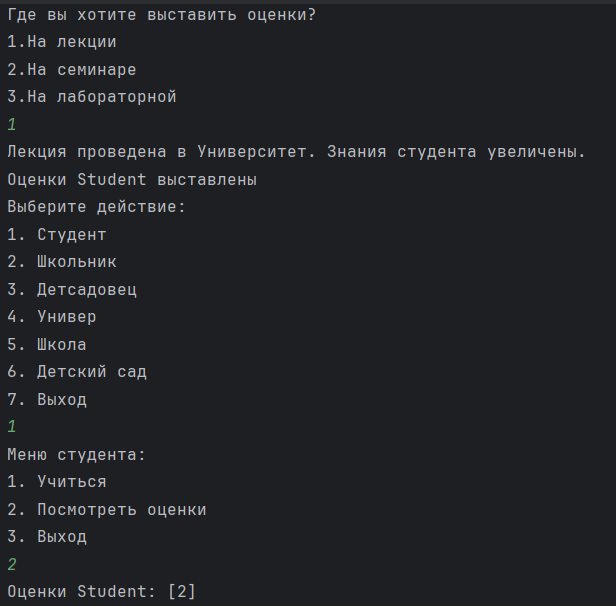


Рисунок 5 – Результат выполнения программы.

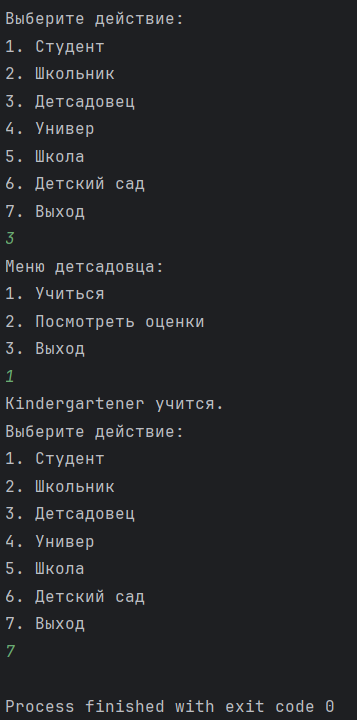


Рисунок 6 – Результат выполнения программы.