

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №1.2
з дисципліни
“ Розроблення клієнтських додатків для мобільних платформ”

Виконала:
студентка групи ІП-83
ЗК 8313
Леут Марія

Київ 2021

Варіант № 2

Скриншоти роботи програми

Частина 1

```
MainKt x
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...

Завдання 1
{ІП-84=[Дмитренко Олександр], ІВ-83=[Бондаренко Максим, Головенець Руслан, Дровнін Павло,
Кочерук Давид, Матвійчук Андрій, Ткаченко Ярослав], ІО-82=[Востриков Нікіта, Лесик Сергій,
Мартинюк Назар, Роман Олександр, Тарасенко Юлія, Фещенко Кирил, Ющенко Андрій],
ІО-83=[Аверкова Анастасія, Крамар Віктор, Соловійов Данііл], ІО-81=[Іванов Володимир, Дудка
Максим, Кобук Назар, Рахуба Вероніка], ІВ-82=[Іванов Дмитро, Лихацька Юлія, Скрипченко
Володимир], ІП-83=[Жуков Михайло, Мінченко Володимир], ІВ-81=[Базова Лідія, Грабко Михайло,
Дрозд Світлана, Кулініч Віталій, Снігурець Олег]}

Завдання 2
{ІП-84={Дмитренко Олександр=[9, 11, 12, 11, 0, 12, 12, 0]}, ІВ-83={Бондаренко Максим=[12, 12,
11, 12, 9, 12, 11, 16], Головенець Руслан=[0, 12, 12, 9, 0, 12, 12, 16], Дровнін Павло=[12,
9, 12, 11, 11, 9, 12, 12], Кочерук Давид=[12, 12, 12, 12, 12, 11, 0], Матвійчук
Андрій=[9, 12, 11, 12, 12, 0, 12, 15], Ткаченко Ярослав=[0, 12, 12, 12, 12, 11, 15]},
ІО-82={Востриков Нікіта=[0, 9, 0, 9, 12, 9, 0, 16], Лесик Сергій=[12, 12, 11, 12, 9, 12, 0,
15], Мартинюк Назар=[11, 12, 12, 12, 12, 0, 11, 16], Роман Олександр=[12, 12, 12, 12, 0, 0,
12, 16], Тарасенко Юлія=[12, 12, 12, 9, 12, 11, 12, 15], Фещенко Кирил=[12, 9, 12, 0, 11,
11, 12, 15], Ющенко Андрій=[9, 12, 0, 11, 12, 11, 12, 15]}, ІО-83={Аверкова Анастасія=[12,
12, 11, 0, 12, 11, 12, 16], Крамар Віктор=[0, 9, 12, 0, 12, 11, 9, 12], Соловійов Данііл=[12,
12, 12, 12, 0, 12, 12, 16]}, ІО-81={Іванов Володимир=[12, 12, 12, 11, 0, 9, 11, 16], Дудка
Максим=[12, 12, 12, 12, 12, 12, 0, 0], Кобук Назар=[9, 12, 0, 12, 0, 12, 9, 15], Рахуба
Вероніка=[12, 12, 11, 12, 0, 0, 9, 16]}, ІВ-82={Іванов Дмитро=[9, 12, 12, 12, 12, 11, 9, 0],
Лихацька Юлія=[0, 0, 12, 0, 12, 9, 12, 0], Скрипченко Володимир=[11, 11, 12, 11, 12, 12,
11, 16]}, ІП-83={Жуков Михайло=[9, 0, 12, 0, 12, 11, 12, 16], Мінченко Володимир=[0, 12, 12,
12, 11, 12, 12, 16]}, ІВ-81={Базова Лідія=[12, 12, 9, 11, 0, 12, 9, 15], Грабко Михайло=[9,
0, 12, 11, 9, 0, 0, 16], Дрозд Світлана=[12, 12, 12, 12, 12, 12, 12, 15], Кулініч
Віталій=[12, 0, 0, 11, 12, 9, 12, 16], Снігурець Олег=[12, 9, 11, 9, 0, 9, 12, 0]}}
```

Завдання 3

```
{IP-84={Дмитренко Олександр=67}, IB-83={Бондаренко Максим=95, Головенець Руслан=73, Дровнін Павло=88, Кочерук Давид=83, Матвійчук Андрій=83, Ткаченко Ярослав=86}, IO-82={Востриков Нікіта=55, Лесик Сергій=83, Мартинюк Назар=86, Роман Олександр=76, Тарасенко Юлія=95, Феценко Кирил=82, Юценко Андрій=82}, IO-83={Аверкова Анастасія=86, Крамар Віктор=65, Соловійов Даниїл=88}, IO-81={Іванов Володимир=83, Дудка Максим=72, Кобук Назар=69, Рахуба Вероніка=72}, IB-82={Іванов Дмитро=77, Лихацька Юлія=45, Скрипченко Володимир=96}, IP-83={Жуков Михайло=72, Мінченко Володимир=87}, IB-81={Базова Лідія=80, Грабко Михайло=57, Дрозд Світлана=99, Кулініч Віталій=72, Снігурець Олег=62}}
```

Завдання 4

```
{IP-84=67.0, IB-83=84.666664, IO-82=79.85714, IO-83=79.666664, IO-81=74.0, IB-82=72.666664, IP-83=79.5, IB-81=74.0}
```

Завдання 5

```
{IP-84=[Дмитренко Олександр], IB-83=[Бондаренко Максим, Головенець Руслан, Дровнін Павло, Кочерук Давид, Матвійчук Андрій, Ткаченко Ярослав], IO-82=[Лесик Сергій, Мартинюк Назар, Роман Олександр, Тарасенко Юлія, Феценко Кирил, Юценко Андрій], IO-83=[Аверкова Анастасія, Крамар Віктор, Соловійов Даниїл], IO-81=[Іванов Володимир, Дудка Максим, Кобук Назар, Рахуба Вероніка], IB-82=[Іванов Дмитро, Скрипченко Володимир], IP-83=[Жуков Михайло, Мінченко Володимир], IB-81=[Базова Лідія, Дрозд Світлана, Кулініч Віталій, Снігурець Олег]}
```

Process finished with exit code 0

Частина 2

CoordinateMLKt ×

"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...

Створення екземпляру координати з нульовими значеннями за замовчуванням

0°0'0" N

Створення екземпляру координати з заданим набором значень (градуси, мінути, секунди)

90°59'5" N

Виведення результату методу, що повертає рядок з градусами, мінутами, секундами

180°14'10" W

Виведення результату методу, що повертає рядок з десятковим значенням координати

180.23611° W

Виведення результату методу координати 5, що повертає середню координату між координатами 5 і 6
10°0'0" S is between 20°0'0" N and 40°0'0" S

Виведення результату методу класу, що повертає середню координату між координатами-параметрами
55°29'32" N is between 90°59'5" N and 20°0'0" N

Process finished with exit code 0

Лістинг коду

Частина 1

Main.kt

```
package ua.kpi.comsys.ip8313.lab1_2

fun main() {
    // Дано рядок у форматі "Student1 - Group1; Student2 - Group2; ..."

    val studentsStr = "Дмитренко Олександр - IP-84; Матвійчук Андрій - IB-83; Лесик Сергій - IO-82; Ткаченко Ярослав - IB-83; Аверкова Анастасія - IO-83;
```

Соловійов Даниїл - ІО-83; Рахуба Вероніка - ІО-81; Кочерук Давид - ІВ-83; Лихацька Юлія - ІВ-82; Головенець Руслан - ІВ-83; Ющенко Андрій - ІО-82; Мінченко Володимир - ІП-83; Мартинюк Назар - ІО-82; Базова Лідія - ІВ-81; Снігурець Олег - ІВ-81; Роман Олександр - ІО-82; Дудка Максим - ІО-81; Кулініч Віталій - ІВ-81; Жуков Михайло - ІП-83; Грабко Михайло - ІВ-81; Іванов Володимир - ІО-81; Востриков Нікіта - ІО-82; Бондаренко Максим - ІВ-83; Скрипченко Володимир - ІВ-82; Кобук Назар - ІО-81; Дровнін Павло - ІВ-83; Тарасенко Юлія - ІО-82; Дрозд Світлана - ІВ-81; Фещенко Кирил - ІО-82; Крамар Віктор - ІО-83; Іванов Дмитро - ІВ-82"

```
// Завдання 1
// Заповніть словник, де:
// - ключ - назва групи
// - значення - відсортований масив студентів, які відносяться до відповідної групи
```

```
val studentsGroups = mutableMapOf<String, List<String>>()
```

```
// Ваш код починається тут
```

```
val studentsList = studentsStr.split("; ")
studentsList.forEach {
    val (name, group) = it.split(" - ")
    val groupStudentsList = (studentsGroups[ group ]?:
mutableListOf<String>()) + name
    studentsGroups[ group ] = groupStudentsList.sorted()
}
```

```
// Ваш код закінчується тут
```

```
println("Завдання 1")
println(studentsGroups)
```

```
// Дано масив з максимально можливими оцінками
```

```
val points = listOf<Int>(12, 12, 12, 12, 12, 12, 12, 16)
```

```
// Заповніть словник, де:
// - ключ - назва групи
// - значення - словник, де:
//   - ключ - студент, який відносяться до відповідної групи
//   - значення - масив з оцінками студента (заповніть масив випадковими
значеннями, використовуючи функцію `randomValue(maxValue: Int) -> Int`)
```

```
fun randomValue(maxValue: Int): Int {
    return when((Math.random() * 6 ).toInt()) {
        1 -> kotlin.math.ceil(maxValue.toFloat() * 0.7).toInt()
        2 -> kotlin.math.ceil(maxValue.toFloat() * 0.9).toInt()
        3, 4, 5 -> maxValue
        else -> 0
    }
}
```

```
val studentPoints = mutableMapOf<String, MutableMap<String, List<Int>>>()
```

```
// Ваш код починається тут
```

```
val groups = studentsGroups.keys;
```

```
for (group in groups) {
```

```

        val studentPointsForGroup = mutableMapOf<String, List<Int>>()
        for (student in studentsGroups[ group ]!!) {
            studentPointsForGroup[ student ] = points.map { randomValue(it) }
        }
        studentPoints[ group ] = studentPointsForGroup
    }

    // Ваш код закінчується тут
    println("Завдання 2")
    println(studentPoints)

    // Завдання 3
    // Заповніть словник, де:
    // - ключ - назва групи
    // - значення - словник, де:
    //   - ключ - студент, який відноситься до відповідної групи
    //   - значення - сума оцінок студента

    val sumPoints = mutableMapOf<String, MutableMap<String, Int>>()

    // Ваш код починається тут

    for (group in groups) {
        val sumPointsForGroup = mutableMapOf<String, Int>()
        val studentPointsForGroup = studentPoints[ group ]!!
        for (student in studentPointsForGroup) {
            val studentName = student.key
            val pointsSumForStudent = studentPointsForGroup[ studentName ]!!.sum()
            sumPointsForGroup[ studentName ] = pointsSumForStudent
        }
        sumPoints[ group ] = sumPointsForGroup
    }

    // Ваш код закінчується тут

    println("Завдання 3")
    println(sumPoints)

    // Завдання 4
    // Заповніть словник, де:
    // - ключ - назва групи
    // - значення - середня оцінка всіх студентів групи

    val groupAvg = mutableMapOf<String, Float>()

    // Ваш код починається тут

    for (group in groups) {
        val sumPointsForGroup = sumPoints[ group ]
        groupAvg[ group ] = sumPointsForGroup!!.values.sum().toFloat() /
(sumPointsForGroup.size)
    }

    // Ваш код закінчується тут

    println("Завдання 4")
    println(groupAvg)

    // Завдання 5

```

```

// Заповніть словник, де:
// - ключ - назва групи
// - значення - масив студентів, які мають >= 60 балів

val passedPerGroup = mutableMapOf<String, List<String>>()

// Ваш код починається тут

for (group in groups) {
    val passedList = mutableList<String>()
    val sumStudentPoints = sumPoints[ group ]
    for (student in sumStudentPoints!!.keys) {
        val grade = sumStudentPoints[ student ]!!
        if (grade >= 60) passedList += student
    }
    passedPerGroup[ group ] = passedList
}

// Ваш код закінчується тут

print("Завдання 5")
print(passedPerGroup)
}

```

Частина 2

CoordinateML.kt

```

package ua.kpi.comsys.ip8313.lab1_2

import java.security.InvalidParameterException
import kotlin.math.abs
import kotlin.math.truncate

enum class Direction(val minDegrees: Int, val maxDegrees: Int,
                    val minMinutes: UInt = 0u, val maxMinutes: UInt = 59u,
                    val minSeconds: UInt = 0u, val maxSeconds: UInt = 59u) {
    LATITUDE(-90, 90),
    LONGITUDE(-180, 180)
}

class CoordinateML(private val direction: Direction) {
    private var degrees: Int = 0
    private var minutes : UInt = 0u
    private var seconds : UInt = 0u
    constructor(
        direction: Direction, degrees: Int = 0, minutes: UInt = 0u, seconds:
UInt = 0u
    ) : this(direction) {
        if (degrees < direction.minDegrees || degrees > direction.maxDegrees) {
            throw InvalidParameterException(
                "Degrees value should be between ${direction.minDegrees} and
${direction.maxDegrees}"
            )
        }
        if (minutes < direction.minMinutes || minutes > direction.maxMinutes) {
            throw InvalidParameterException(
                "Minutes value should be between ${direction.minMinutes} and

```

```

    "${direction.maxMinutes}"
    )
    }
    if (seconds < direction.minSeconds || seconds > direction.maxSeconds) {
        throw InvalidParameterException(
            "Seconds value should be between ${direction.minSeconds} and
    }
    }
    this.degrees = degrees
    this.minutes = minutes
    this.seconds = seconds
}

private fun getDirection(): String {
    return when (this.direction) {
        Direction.LATITUDE -> {
            if (this.degrees >= 0) "N"
            else "S"
        }
        Direction.LONGITUDE -> {
            if (this.degrees >= 0) "E"
            else "W"
        }
    }
}

private fun getDecimal(): Float {
    val minsAndSecs = (this.minutes.toFloat() + this.seconds.toFloat() / 60) /
60
    return if (this.degrees > 0) this.degrees.toFloat() + minsAndSecs
        else this.degrees.toFloat() - minsAndSecs
}

fun getValueForPrint(): String {
    val dir = this.getDirection()
    return "${abs(degrees)}°${minutes}'${seconds}" $dir"
}

fun getDecimalForPrint(): String {
    return "${abs(this.getDecimal())}° ${this.getDirection()}"
}

companion object {
    @JvmStatic
    private fun revertFromDecimal(decimalVal: Float): Triple<Int, UInt, UInt>
{
        val degrees = truncate(decimalVal).toInt()
        val minsAndSecs = abs((decimalVal - degrees) * 60)
        val minutes = truncate(minsAndSecs).toUInt()
        val seconds = truncate((minsAndSecs - minutes.toDouble()) *
60).toUInt()
        return Triple(degrees, minutes, seconds)
    }
    @JvmStatic
    fun getHalfwayCoordBetween(start: CoordinateML, end: CoordinateML):
CoordinateML? {
        if (start.direction != end.direction) return null
        val halfway = (start.getDecimal() + end.getDecimal()) / 2
    }
}

```

```

        val (halfwayDegrees, halfwayMinutes, halfwaySeconds) =
CoordinateML.revertFromDecimal(halfway)
        return CoordinateML(start.direction, halfwayDegrees, halfwayMinutes,
halfwaySeconds)
    }
}

fun getHalfwayCoordTo(coord: CoordinateML): CoordinateML? {
    if (this.direction != coord.direction) return null
    val start = this.getDecimal()
    val end = coord.getDecimal()
    val halfway = (start + end) / 2
    val (halfwayDegrees, halfwayMinutes, halfwaySeconds) =
CoordinateML.revertFromDecimal(halfway)
    return CoordinateML(this.direction, halfwayDegrees, halfwayMinutes,
halfwaySeconds)
}

fun main() {
    println("Створення екземпляру координати з нульовими значеннями за
замовчуванням")
    val coord1 = CoordinateML(Direction.LATITUDE)
    println(coord1.getValueForPrint())
    println("Створення екземпляру координати з заданим набором значень (градуси,
мінути, секунди)")
    val coord2 = CoordinateML(Direction.LATITUDE, 90, 59u, 5u)
    println(coord2.getValueForPrint())
    val coord3 = CoordinateML(Direction.LONGITUDE, -180, 14u, 10u)
    println("Виведення результату методу, що повертає рядок з градусами, мінутами,
секундами")
    println(coord3.getValueForPrint())
    println("Виведення результату методу, що повертає рядок з десятковим значенням
координати")
    println(coord3.getDecimalForPrint())
    val coord4 = CoordinateML(Direction.LATITUDE, 20, 0u, 0u)
    val coord5 = CoordinateML(Direction.LATITUDE, -40, 0u, 0u)
    val coord6 = coord4.getHalfwayCoordTo(coord5)
    println("Виведення результату методу координати 5, що повертає середню
координату між координатами 5 і 6")
    println("${coord6?.getValueForPrint()} is between ${coord4.getValueForPrint()}
and ${coord5.getValueForPrint()}")
    println("Виведення результату методу класу, що повертає середню координату між
координатами-параметрами")
    val coord7 = CoordinateML.getHalfwayCoordBetween(coord2, coord4)
    println("${coord7?.getValueForPrint()} is between ${coord2.getValueForPrint()}
and ${coord4.getValueForPrint()}")
}

```

Висновок

Під час виконання лабораторної роботи познайомилася з основами мови Kotlin, її засобами для роботи з рядками, числами, структурами даних, навчилася створювати списки, словники, класи, дізналася про різні способи ініціалізації, навчилася оголошувати властивості та методи, зокрема статичні. Створила дві

прості програми, що дають змогу ознайомитися з основами мови. Мету роботи виконано.