

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

КУРСОВАЯ РАБОТА
по дисциплине «Объектно-ориентированное программирование»
Тема: Приложение «Телефонная книга»

Студент гр. 8302

Халитов Ю.Р.

Преподаватель

Красильников А.В.

Санкт-Петербург

2021

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент Халитов Юлиан Ришелевич

Группа 8302

Тема работы: Приложение «Телефонная книга»

Исходные данные:

Приложения для просмотра и редактирования информации контактов телефонной книги.

Содержание пояснительной записки:

«Введение», «Используемые технологии», «UML диаграммы», «Use-Case диаграмма», «Основные элементы исходного кода», «Выводы».

Предполагаемый объем пояснительной записки:

Не менее 5 страниц.

Дата выдачи задания: 18.11.2020

Дата сдачи курсовой работы : 06.03.2021

Дата защиты курсовой работы: 06.03.2021

Студент

Халитов Ю.Р.

Преподаватель

Красильников А.В.

АННОТАЦИЯ

Целью выполнения курсового проекта является разработка приложения с графическим интерфейсом. Курсовой проект обобщает все знания и навыки, полученные во время изучения дисциплины.

SUMMARY

The purpose of the course project is to develop an application with a graphical interface. The course project summarizes all the knowledge and skills acquired during the study of the discipline.

ВВЕДЕНИЕ

Необходимо разработать приложение для просмотра и редактирования информации контактов телефонной книги.

У каждого контакта может быть следующая информация

- ФИО
- Номера телефонов
- Email'ы
- Адрес
- Дата рождения

Функциональность:

- Создание нового контакта
- Удаление существующего
- Редактирование информации о контакте
- Экспортирование информации о контакте в формате vCard-файла

ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

Язык программирования: Kotlin

Система сборки: Gradle

Сторонние библиотеки:

1. ezvcard – библиотека предоставляет классы для работы с форматом vCard, а также методы сериализации/десериализации
2. tornadofx – библиотека для создания графического интерфейса программы

UML-ДИАГРАММЫ

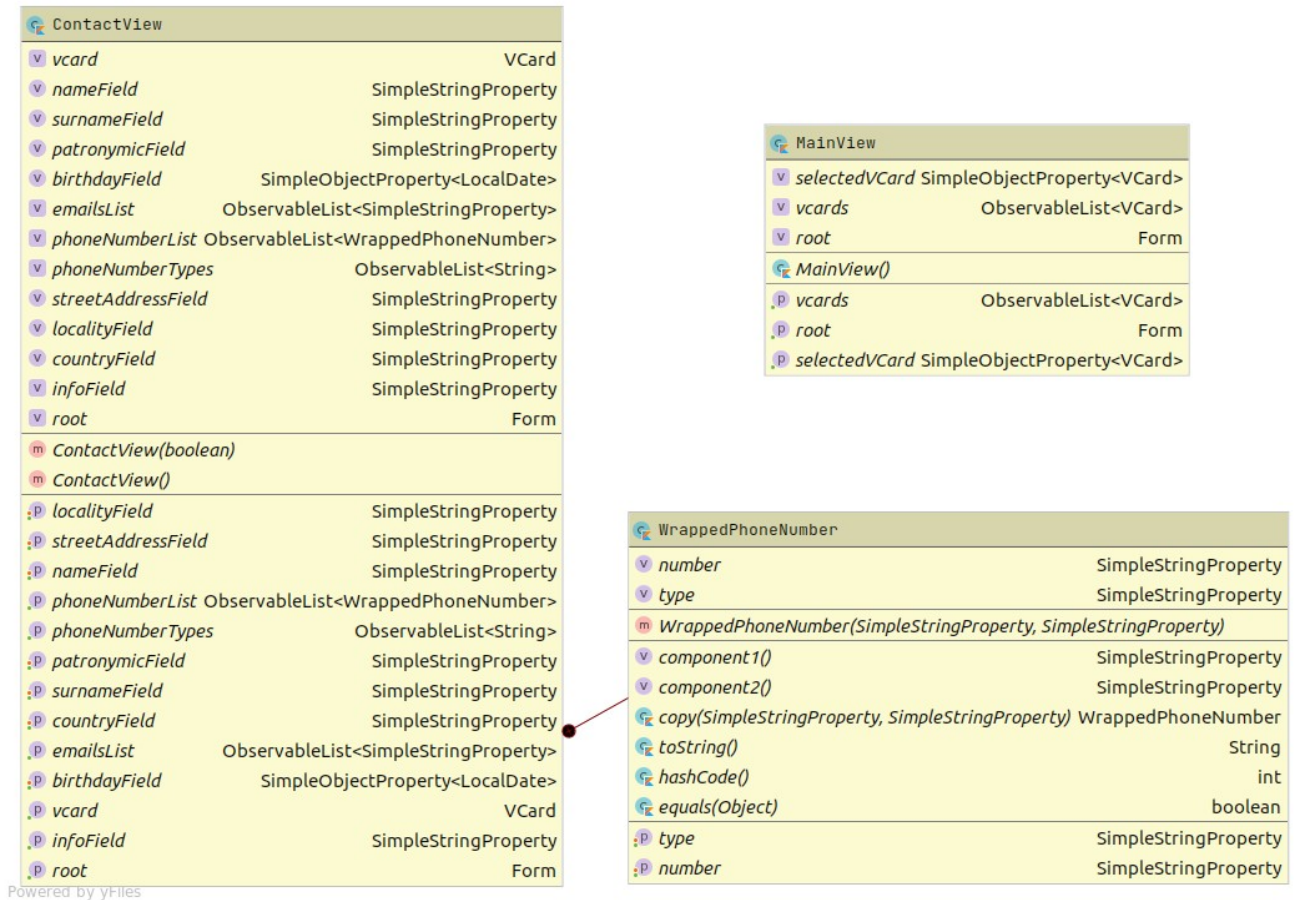


Рис. 1 — UML-диаграмма пакета view

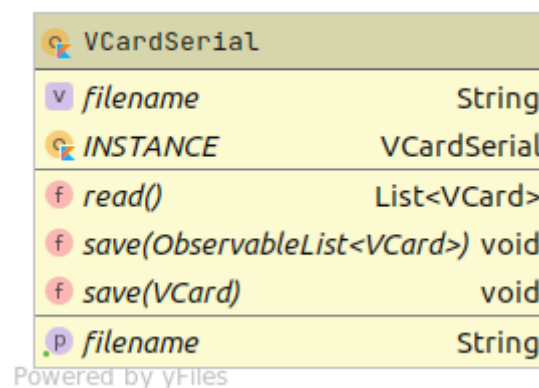


Рис. 2 — UML-диаграмма пакета vcards

USE-CASE ДИАГРАММА

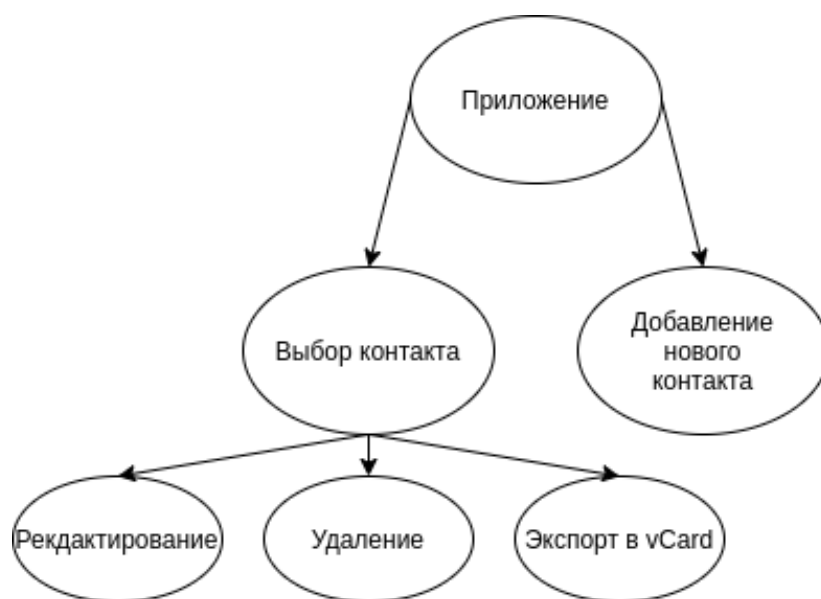


Рис. 3 — Use-Case диаграмма приложения

ОСНОВНЫЕ ЭЛЕМЕНТЫ ИСХОДНОГО КОДА

```
object VCardSerial {
    val filename = "vcards.vcf"
    fun read() : List<VCard> {
        val vcards = mutableListOf<VCard>()
        val file = File(filename)
        if (file.exists()) {
            val reader = VCardReader(file)
            reader.use { reader ->
                while (true) {
                    val vcard = reader.readNext() ?: break
                    vcards.add(vcard)
                }
            }
        }
        return vcards
    }

    fun save(vcards: ObservableList<VCard>) {
        val file = File(filename);
        val writer = VCardWriter(file, VCardVersion.V4_0)
        writer.use {
            vcards.forEach { vcard ->
                it.write(vcard)
            }
        }
    }

    fun save(vcard: VCard) {
        val name = vcard.structuredName.given
        val filename = name + ".vcf"
        var file = File(filename)
        var i = 0
        while (file.exists()) {
            file = File("$name${i++}.vcf")
        }
        val writer = VCardWriter(file, VCardVersion.V4_0)
        writer.use {
            it.write(vcard)
        }
    }
}

class MainView : View("Contacts App") {

    val selectedVCard = SimpleObjectProperty<VCard>()
    val vcards = FXCollections.observableArrayList(VCardSerial.read())

    override val root = form {
        listview(vcards) {
            prefWidth = 400.0
            cellFormat { vcard ->
                graphic = form {
                    fieldset {
                        label {
                            style {
                                fontSize = 22.px
                            }
                        }
                    }
                }
            }
        }
    }
}
```



```

        fontWeight = FontWeight.BOLD
    }
    val n = vcard.structuredName
    text = "${n?.given ? : ""} ${n?.family ? : ""} $
{n?.additionalNames?.firstOrNull() ? : ""}"
    }
    field("Phone Numbers") {
        label {
            vcard.telephoneNumbers.forEach {
                text += "${it.text ? : ""} ($
{it.types.firstOrNull()?.value ? : ""})\n"
            }
        }
    }
    field("Emails") {
        label {
            vcard.emails.forEach {
                it.value?.let {
                    text += it + '\n'
                }
            }
        }
    }
    field("Birthday") {
        label {
            val format = SimpleDateFormat("dd/MM/yyyy")
            text = vcard.birthday?.let
{ format.format(it.date) }
        }
    }
    fieldset("Address") {
        field("Street") {
            label {
                text =
vcard.addresses?.firstOrNull()?.streetAddress
            }
        }
        field("Locality") {
            label {
                text = vcard.addresses?.firstOrNull()?.locality
            }
        }
        field("Country") {
            label {
                text = vcard.addresses?.firstOrNull()?.country
            }
        }
    }
    }
    }
    bindSelected(selectedVCard)
}
button("New") {
    useMaxWidth = true
    action {
        ContactView().openWindow()
    }
}
button("Edit") {
    useMaxWidth = true
    action {
        val vcard = selectedVCard.value

```

```

        val index = vcards.indexOf(vcard)
        if (index >= 0 && index < vcards.size) {
            ContactView(is_new = false).openWindow()
        }
    }
}
button("Delete") {
    useMaxWidth = true
    action {
        val vcard = selectedVCard.value
        val index = vcards.indexOf(vcard)
        if (index >= 0 && index < vcards.size) {
            vcards.remove(vcard)
            VCardSerial.save(vcards)
        }
    }
}
button("Import VCard") {
    useMaxWidth = true
    action {
        val vcard = selectedVCard.value
        val index = vcards.indexOf(vcard)
        if (index >= 0 && index < vcards.size) {
            VCardSerial.save(vcard)
        }
    }
}
}
}
}
}

```

ВЫВОДЫ

В ходе выполнения курсовой работы было разработано приложение с графическим интерфейсом «Телефонная книга», получен опыт работы с языком Kotlin, системой сборки Gradle, использования сторонних библиотек.