

Министерство науки и высшего образования Российской Федерации

Томский государственный университет
систем управления и радиоэлектроники

И.В. Безходарнов

РАЗРАБОТКА ПРИЛОЖЕНИЙ

Методические указания
для проведения практических занятий

Томск 2025

УДК 004.021
ББК 32.973.4
Б39

Рецензент:

Гриценко Ю.Б., кафедра АОИ ТУСУР, к.т.н., доцент

Автор: И.В. Безходарнов

Безходарнов, Илья Владимирович

Б39 Разработка приложений: метод. указания для проведения практических занятий / И.В. Безходарнов – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2025. – 14 с.

Методические указания предназначены для проведения практических занятий по дисциплине "Разработка приложений". В ходе занятий предлагается на учебных примерах изучить и получить практические навыки работы с несколькими популярными технологиями для создания приложений и общими принципами их построения.

Одобрено на заседании ПИШ, протокол № 9 от 03.09.2024 г.

УДК 004.021
ББК 32.973.4

© Безходарнов И.В., 2025

© Томск. гос. ун-т систем упр. и радиоэлектроники, 2025

Оглавление

Введение.....	4
Тема №1 Инструменты для разработки приложений.....	5
Тема №2 Разработка приложений с графическим интерфейсом на языке C# и Python8	
Тема №3 Разработка кроссплатформенных приложений с использованием технологии Electron.....	11
Заключение	13
Список рекомендуемой литературы.....	14

Введение

Разработка приложений – это уже давно развивающаяся отрасль, в которой постоянно возникают новые технологии, подходы, методологии. Изначально технологически сложная ИТ-область, требующая глубокого понимания всех процессов в вычислительных системах, системных знаний в области программирования, постепенно ищет пути, для снижения вложений в получение результата. Современные технологии создания приложений имеют тренд скрывать за простотой и интуитивной понятностью все сложности взаимодействия с вычислительной системой, оставляют разработчику в основном творчество в решении прикладных задач. А технологии ИИ, в частности большие языковые модели позволяют автоматизировано создавать исходный код для решения широкого круга типовых задач.

В процессе выполнения практических занятий по данной дисциплине вам будет предложено создать два учебных приложения с использованием различных технологий, а также два небольших «модельных» приложения для ознакомления с еще двумя технологиями для создания приложений. В процессе выполнения задач можно использовать помощь больших языковых моделей, документацию, консультации преподавателя.

Результатом выполнения практических занятий станут приобретенные знания и навыки практической работы, которые можно будет использовать как стартовую площадку для дальнейшего освоения данной области ИТ-индустрии или помогут в дальнейшем осознанно ориентироваться в вопросах создания приложений для организации работы над комплексными проектами, в которые входит и программное и аппаратное обеспечение.

Тема №1 Инструменты для разработки приложений

Цель: научиться устанавливать, настраивать и использовать интегрированную среду разработки на примере Microsoft VS Code.

Задачи:

1. Ознакомиться с продуктом Microsoft VS Code изучив обзорный раздел из официальной документации.
2. Установить и настроить VS Code для работы с языком Python и системой контроля версий git.
3. Создать учебное приложение и освоить на практике работу в интегрированной среде разработки.

Теоретическая часть

Интегрированная среда разработки (IDE – Integrated Development Environment) – это набор инструментов, которые позволяют писать исходный код, компилировать его, отлаживать приложения через единый интерфейс. Примеры таких продуктов:

- Microsoft Visual Studio Code (<https://code.visualstudio.com>);
- Microsoft Visual Studio (<https://visualstudio.microsoft.com>);
- семейство IDE от компании JetBrains (<https://www.jetbrains.com/ides>);
- Android Studio (<https://developer.android.com>).

Интегрированная среда разработки делает труд разработчика более эффективным, предоставляя ему различные инструменты, ускоряющие получение результата, которыми он может пользоваться через единый интерфейс. Примеры инструментов-помощников:

- подсветка синтаксиса исходного кода;
- предварительная компиляция кода и подсветка ошибок;
- in-line подсказки для написания кода, автоматическая подстановка или справка по вариантам написания директив языка;
- компиляция и сборка проекта «в один клик»;
- встроенный отладчик (Debugger);
- встроенный в графический интерфейс помощник работы с системой контроля версий;
- инструменты для сравнения версий файлов с исходным кодом, разрешения конфликтов при синхронизации версий с репозитарием;
- средства форматирования исходного кода по заданным правилам;
- интеллектуальные помощники написания и проверки кода;
- инструменты типа командная-строка, терминал и др.

Если разработчик не использует IDE, то он вынужден:

- писать исходный код в обычном редакторе, без средств предкомпиляции и автоподсказок;
- запускать средства компиляции и сборки вручную из командной строки;

- отлаживать исходный код путем вставки в него специальных директив логирования и читать их из файла или консоли;
- работать с системой контроля версий через командную строку.

Иногда, приходится применять методы разработки без IDE, например, когда нужно провести отладку или разобраться с ошибками, которые возникают только на удаленных серверах, или на компьютерах пользователя, где нет возможности установить и настроить IDE. И такие действия обычно занимают много времени и сложны организационно.

Для данных практически занятий выбрана свободно распространяемая IDE с открытым исходным кодом от компании Microsoft: Visual Studio Code (<https://code.visualstudio.com>). Она написана с помощью технологии Electron, является кроссплатформенной, может работать под операционными системами Windows, Linux и macOS. Для ее установки не требуются права администратора. С помощью дополнительных инструментов, которые устанавливаются в добавок к студии она может работать со многими языками программирования.

Практическая часть

Для выполнения первой задачи данного практического занятия ознакомьтесь с обзором возможностей продукта на официальной странице документации: <https://code.visualstudio.com/docs>. Далее, для базового изучения, скачайте и установите приложение, затем пройдите по шагам ознакомительное руководство: <https://code.visualstudio.com/docs/getstarted/getting-started>.

Вторая задача – настройка среды для работы с языком программирования Python. Воспользуйтесь официальной документацией: <https://code.visualstudio.com/docs/python/python-quick-start>, пройдите ее по шагам. В результате у вас получится простейшее приложение в одном файле исходного кода. Чтобы завершить задачу – нужно инициализировать локальный git репозиторий и сохранить в него полученный файл. Для этого пройдите по шагам ознакомительное руководство по работе с системой контроля версий в VS Code: <https://code.visualstudio.com/docs/sourcecontrol/overview>. Затем инициализируйте локальный git репозиторий для этого проекта, выполните первый коммит для сохранения полученного в ходе выполнения задачи файла исходного кода на языке Python.

Третья задача – создание учебного приложения на языке Python, работающего в командной строке по одному из предлагаемых вариантов заданий:

1. Простейший архиватор, без сжатия. В командной строке задаются: команда, выполнить архивацию или выполнить разархивацию; файл архива, в который будут записаны все нужные файлы или разархивированы из него, и сами файлы для архивации.

2. Поиск текста по всем файлам, находящимся в каталоге. В командной строке задаются текст для поиска, с возможностью вводить текст, содержащий пробелы и маска для поиска файлов, например, *.txt.

3. Вычисление контрольной суммы файла по трем различным алгоритмам, выбрать которые следует самостоятельно. В командной строке задаются алгоритм вычисления контрольной суммы и имя файла. Результат вычисления выводится в консоль.

4. Приложение по собственному предложению, которое нужно предварительно согласовать с преподавателем.

Для каждого из вариантов должна быть реализована встроенная документация – запуск приложения с флагом -h и/или --help должны выводить в консоль справку по использованию.

Для завершения выполнения практического занятия нужно инициализировать локальный git репозиторий, залить в него исходные коды приложения и затем опубликовать этот репозиторий в одном из открытых хранилищ со свободным доступом, например github или любом другом аналоге.

Полученные результаты

После выполнения практического задания:

1. У вас должны сформироваться знания и навыки работы с IDE Microsoft Visual Studio Code. Вспомните и повторите ее основные особенности: функции, работа с разными языками, основные элементы интерфейса.

2. Работающее приложение по одному из вариантов задания, убедитесь, что оно функционирует без ошибок во всех заявленных режимах.

3. Репозиторий с публичным доступом, в котором находится исходный код вашего приложения.

Тема №2 Разработка приложений с графическим интерфейсом на языке C# и Python

Цель: разработать модельное приложение с графическим интерфейсом.

Задачи:

1. Настроить интегрированную среду разработки VS Code для работы над проектами на языке C#.
2. Создать модельное приложение на языке C#, добиться его работоспособности.
3. Создать модельное приложение на языке Python, добиться его работоспособности.

Теоретическая часть

Разработка приложений для различных операционных систем может выполняться на разных технологиях. Ключевыми различиями технологий являются:

1. Сложность освоения и использования самой технологии.
2. Насколько конкретная технология подходит для конкретной операционной системы.
3. Возможность, сложность, стоимость создания кроссплатформенных приложений на этой технологии.
4. Возможность использования системных ресурсов конкретной операционной системы для выбранной технологии создания приложений.
5. Ресурсоемкость приложений, созданных на основе конкретной технологии.

В ходе данного практического занятия нужно создать модельное приложение с графическим интерфейсом, имеющее простейший вид на двух популярных языках программирования: C# и Python, разобраться каким образом они могут поставляться пользователю, сделать выводы о применимости этих технологий для разных операционных систем.

Практическая часть

Для выполнения задачи 1 можно воспользоваться официальной инструкцией по установке расширений и настройке VS Code для работы с языком C#: <https://code.visualstudio.com/docs/csharp/get-started>.

Прежде чем делать приложение с графическим пользовательским интерфейсом, создайте простейшее консольное приложение типа «Hello World».

Далее нужно разобраться, как создать приложение на C# с графическим пользовательским интерфейсом (GUI – Graphics User Interface). Сам инструмент VS Code не предназначен для создания и разработки таких приложений, поэтому нужно воспользоваться возможностями платформы .NET для создания приложений, которые доступны через командную строку (вне среды IDE Visual Studio Code).

Примечание: для создания проекта C#, использующего библиотеку WinForms, в текущем каталоге можно воспользоваться командой «dotnet new winforms –n .».

Для завершения второго пункта практического занятия достаточно создать простейшее приложение, например такое:

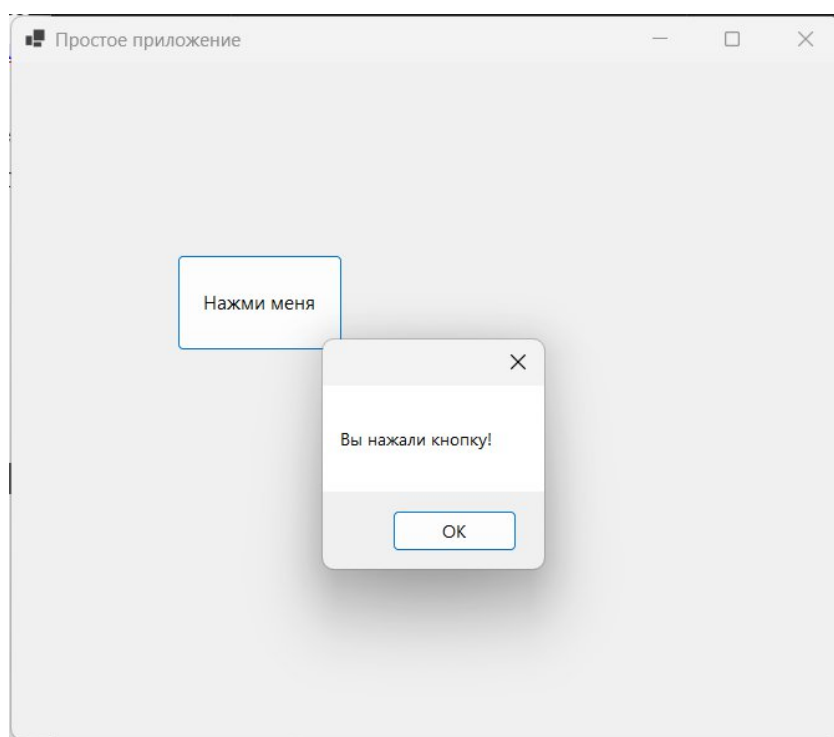


Рисунок 2.1

Для выполнения третьей задачи практического занятия достаточно будет воспользоваться одной из графических библиотек для Python, например Tkinter или подобной. Здесь также достаточно создать простейшее приложение, аналогичное предыдущему пункту. Дополнительно следует разобраться, каким образом можно передать первое и второе приложение пользователю, чтобы он мог запустить его на своем компьютере. Кроме того, нужно ответить на вопрос, будет ли являться хотя бы одно из этих приложений кроссплатформенным.

Полученные результаты

1. Исходный код и приложение для Windows, написанное на языке C#, описание как это приложение можно передать пользователю.
2. Исходный код на языке Python, описание как можно передать это пользователю, чтобы он мог запустить его на своем компьютере. В описании сделайте уточнение на каких ОС это можно сделать.

Тема №3 Разработка кроссплатформенных приложений с использованием технологии Electron

Цель: ознакомиться на практике с разработкой кроссплатформенных приложений с применением технологии Electron.

Задачи:

1. Настроить среду разработки для работы и отладки с Electron, создать стартовое приложение.
2. Создать учебное приложение с графическим интерфейсом.
3. Залить код учебного приложения в публичных git репозиторий.

Теоретическая часть

В ходе данного занятия предлагается разработать учебное приложение, которое реализует графический интерфейс для сделанного на первом занятии приложения, работающего через командную строку.

GUI приложение дает возможность пользователю задать параметры интерактивно, через поля ввода, затем запускает дочерний процесс системы – приложение командной строки, дожидается его выполнения, считывает статус завершения и выдает пользователю результат операции – успех, или ошибку.

GUI приложение предлагается разработать с использованием технологии Electron: <https://www.electronjs.org>, работая при этом в IDE Microsoft Visual Studio Code: <https://code.visualstudio.com>. Для этого нужно будет сначала настроить среду для работы с Electron, затем создать приложение и сохранить исходный код на git репозиторий с публичным доступом.

Практическая часть

Для выполнения задачи №1 достаточно скачать шаблон Electron приложения и настроить конфигурацию для его отладки. Для подсказки можно воспользоваться инструкцией с официального сайта: <https://www.electronjs.org/docs/latest/tutorial/debugging-vscode>. VS code имеет встроенную поддержку приложений на JavaScript и Node.js, поэтому дополнительной установки расширений не требуется. Единственное, для работы темплейта приложения на Electron нужно будет установить необходимые библиотеки в ваш проект, например, такой командой: «npm install electron --save-dev».

В скачанном темплейте приложения реализовано создание окна с простейшим содержимым, и в случае правильной настройки запущенное приложение будет выглядеть примерно так:

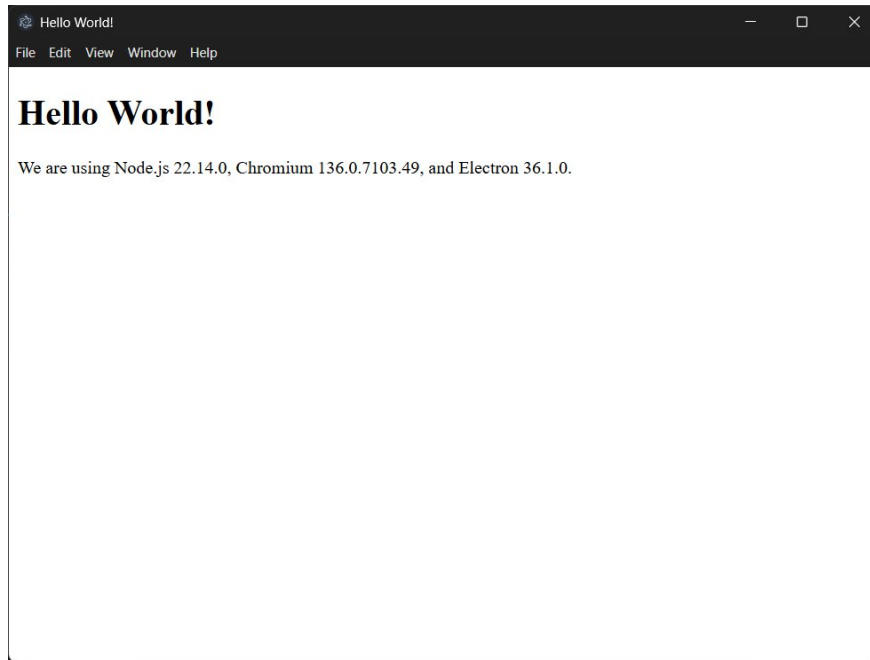


Рисунок 3.1

Для выполнения второй задачи практического занятия нужно воспользоваться документацией на Electron: <https://www.electronjs.org/docs/latest>, помощью искусственного интеллекта и собственными знаниями в области разработки на JavaScript, а также консультациями преподавателя. К приложению не предъявляется требований по внешнему виду, хотя аккуратный дизайн предпочтительнее. К исходному коду предъявляется следующее требование – вы должны быть способны пояснить значение всех блоков и каждой строчки исходного кода, т.е. должны разобраться в его смысле и назначении даже если делаете задание по подсказке, данной нейронной сетью. Кроме того, форматирование исходного кода должно быть аккуратным, а его структура максимально прозрачной, простой для восприятия.

Для выполнения третьей задачи данного занятия воспользуйтесь одним из публичных, бесплатных сервисов хранения и публикации исходных кодов под управлением git. Ссылку на публичный репозиторий, в котором находится исходный код вашего приложения предоставьте для проверки.

Полученные результаты

В результате третьего занятия должны получиться следующие артефакты:

1. Настроенная для работы с технологией Electron среда разработки VS Code.
2. Работющее приложение с графическим пользовательским интерфейсом.
3. Исходный код приложения, сохраненный в публичный git репозиторий.

Заключение

В ходе выполнения заданий, представленных в пройденных практических занятиях, вы ознакомились с тремя языками программирования для создания приложений для локального использования. Для получения дополнительных знаний можно самостоятельно изучить способы оформления приложений, написанных на данных языках, для установки на компьютер пользователя в виде установочного пакета. Поняв отличия между ними, вы сможете делать выбор в сторону той или иной технологии разработки в том числе, и в зависимости от данного критерия.

Кроме языков программирования, были получены навыки практической работы, установки, настройки интегрированной среды разработки Microsoft Visual Studio Code, и вы можете в дальнейшем, при необходимости использовать данные навыки в полной мере, они достаточны для того, чтобы самостоятельно сконфигурировать IDE для работы с другими языками и технологиями программирования.

Публичный репозиторий с исходным кодом, написанного вами приложения может служить примером вашей работы, которым можно поделиться для представления вашего пусть небольшого, но реального опыта и навыков работы по созданию приложений.

Список рекомендуемой литературы

1. Microsoft Visual Studio Code : сайт / microsoft. – URL: <https://code.visualstudio.com>.
Режим доступа: свободный (дата обращения 15.08.2024).
2. Microsoft Visual Studio : сайт / microsoft. – URL: <https://visualstudio.microsoft.com>.
Режим доступа: свободный (дата обращения 15.08.2024).
3. Семейство IDE от компании : сайт / JetBrains. – URL <https://www.jetbrains.com/ides>.
Режим доступа: свободный (дата обращения 15.08.2024).
4. Android Studio : сайт / android. – URL <https://developer.android.com>. Режим доступа свободный (дата обращения 15.08.2024).
5. Технология Electron : сайт / OpenJS. – URL: <https://www.electronjs.org>. Режим доступа свободный (дата обращения 15.08.2024).