**ТИТУЛЬНЫЙ ЛИСТ**

**Содержание**

|  |  |
| --- | --- |
| Введение |  |
| 1. Теоретические аспекты разработки оконных приложений |  |
| * 1. Этапы развития оконных приложений |  |
| * 1. Основные языки программирования и их свойства, необходимые для разработки оконных приложений |  |
| 1. Разработка оконного приложения фоторедактора |  |
| * 1. Этапы жизненного цикла разработки программного обеспечения на примере оконного приложения |  |
| * 1. Процесс разработки оконного приложения фоторедактора |  |
| Заключение |  |
| Список использованных источников |  |
| Приложения |  |

**Введение**

В данном исследовании рассмотрены разновидности оконных приложений, а также будет представлен алгоритм разработки программного обеспечения на примере оконного приложения для обрабатывания фотографий.

Целью выпускной квалификационной работы является разработка детского фоторедактора на языке Python.

Исходя из цели работы были поставлены следующие задачи:

* Анализ предметной области;
* Выбор технологий, языка и среды разработки
* Разработка и оптимизация алгоритма работы оконного приложения
* Выбор Фреймворка

Объектом исследования является технология разработки оконных приложений.

Актуальность выпускной квалификационной работы обусловлена высокой популярностью оконных приложений, в том числе и фоторедактора, для начинающих разработчиков программного обеспечения или дизайнеров программного обеспечения, так как первым и вторым позволяют быстро получить требуемую информацию о разработки ПО.

В современном информационном обществе оконные приложения занимают центральное место в мире программирования. С каждым годом они становятся все более популярными и востребованными. Сейчас мы рассмотрим причины, почему оконные приложения так популярны в 21 веке, их практическую значимость и влияние на различные сферы деятельности.

Удобство использования:

Одной из главных причин популярности оконных приложений является их удобство использования. Окна, кнопки, меню и другие элементы интерфейса делают приложения интуитивно понятными даже для непрофессионалов. Благодаря этому, пользователи могут легко взаимодействовать с приложениями и выполнять необходимые задачи без особых усилий.

Кросс-платформенность:

Оконные приложения обладают кросс-платформенной совместимостью, что позволяет им работать на различных операционных системах, таких как Windows, macOS и Linux. Это делает их универсальными и доступными для широкой аудитории пользователей. Разработчики могут создавать приложения, которые работают на разных платформах, минимизируя затраты и увеличивая охват пользователей.

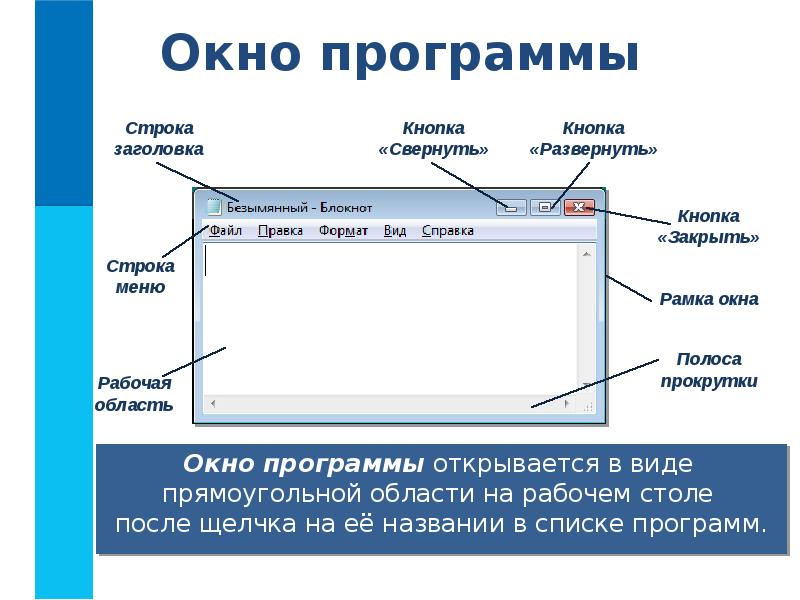
Расширенные возможности:

Оконные приложения предоставляют разработчикам широкий набор инструментов и библиотек для реализации различных функциональностей. С помощью графического интерфейса и множества доступных API, разработчики могут создавать приложения с продвинутыми возможностями, такими как обработка изображений, работа с базами данных, сетевое взаимодействие и многое другое. Это позволяет создавать мощные и гибкие приложения, которые отвечают различным потребностям пользователей.

Работа состоит из двух глав, теоретической и практической части, выполнена на – \_\_\_ страницах, содержит \_\_ рисунков, \_\_ таблиц.

1. **Теоретические аспекты разработки оконных приложений**
   1. **Этапы развития оконных приложений**

В современном мире разработка программного обеспечения превратилась в одну из самых дорогостоящих индустрий и любые узкие места в технологическом процессе его создания могут привести к нежелательным результатам. Удлинение сроков разработки программного обеспечения чревато удорожанием конечного продукта, а не выявленные в ходе тестирования ошибки приводят как минимум к снижению его производительности. Примитивные ошибки, невнятные сообщения и неряшливый интерфейс раздражают пользователей, которые в итоге выбирают более качественный продукт конкурента, а фирма рискует потерять не только клиентов, но и свою долю рынка. Поэтому, качество программного обеспечения приобретает первостепенное значение. Разберём детальнее один из аспектов программного обеспечения-оконные приложения.



История создания оконных приложений включает множество компаний и инженеров, которые работали над различными аспектами этой технологии на протяжении многих лет. Одной из первых компаний, которая начала работу над оконными приложениями, была Xerox PARC (Palo Alto Research Center), куда в 1970-х годах перешли многие ученые, работавшие в лаборатории исследования и развития компании Xerox.

В Xerox PARC были разработаны многие новые технологии, включая графический пользовательский интерфейс (GUI) и сенсорный экран. Одним из самых известных изобретений, созданных в Xerox PARC, был концепт оконного интерфейса, который позволял пользователям работать с несколькими приложениями одновременно в разных окнах.

Позже компания Apple привлекла некоторых инженеров из Xerox PARC, чтобы помочь им создать графический пользовательский интерфейс для своих компьютеров Macintosh, выпущенных в 1984 году. Этот интерфейс был основан на концепции оконного интерфейса, и был назван "Finder".

С развитием оконных приложений возникла необходимость в более сложных приложениях, таких как графические редакторы. Один из наиболее известных графических редакторов - Adobe Photoshop - был создан в 1987 году братьями Джоном и Томасом Кноллами. Они создали прототип приложения на языке программирования Pascal на компьютере Apple Macintosh, а затем переписали его на языке С, чтобы повысить производительность.

С тех пор оконные приложения стали неотъемлемой частью компьютерной технологии, и появились многие другие графические редакторы и приложения для работы с мультимедийными файлами. Сегодня они являются ключевой частью нашей повседневной жизни и широко используются в различных отраслях, включая дизайн, медицину, науку и образование.

Со временем появились новые операционные системы, и каждая из них имела свои собственные интерфейсы и стандарты для создания приложений. Например, в 1984 году Apple выпустила Macintosh с операционной системой MacOS и графическим интерфейсом пользователя (GUI) с использованием мыши.

Это стало большим шагом вперед для разработки оконных приложений. Оконные приложения теперь стали более доступными и удобными для пользователей. Вскоре после этого Microsoft выпустила операционную систему Windows, которая стала популярной и широко распространенной, благодаря чему стандарты оконного интерфейса пользователя были установлены.

Создание программного обеспечения, включая оконные приложения и Photoshop, требует огромного количества времени, усилий и технических знаний. Но благодаря усилиям разработчиков, мы можем пользоваться множеством приложений и программ, которые значительно упрощают нашу жизнь и делают нашу работу более продуктивной.

Создание оконных приложений имеет множество преимуществ. Некоторые из них:

Интуитивно-понятный интерфейс: Оконные приложения имеют привычный и удобный интерфейс, который позволяет пользователям легко взаимодействовать с приложением.

Удобство использования: Оконные приложения позволяют работать с несколькими приложениями одновременно, что повышает эффективность работы.

Большой функционал: Оконные приложения обеспечивают доступ к большому количеству функций, таких как редактирование текста, работа с изображениями и звуком, обработка данных и т.д.

Повышенная производительность: Оконные приложения обычно работают быстрее и более стабильно, чем веб-приложения или приложения, работающие в командной строке.

Широкий выбор инструментов разработки: Существует множество инструментов разработки для создания оконных приложений, таких как Visual Studio, Qt Creator, Eclipse и многие другие, что позволяет разработчикам выбирать наиболее подходящий для них инструмент.

Возможность локального хранения данных: Оконные приложения могут хранить данные локально, что может быть полезно в случаях, когда интернет-соединение недоступно или нестабильно.

Оконные приложения можно создавать на разных языках программирования, в зависимости от конкретных потребностей и предпочтений разработчика. Вот несколько популярных языков, которые широко используются для создания мощных оконных приложений:

C++: C++ является одним из самых популярных языков для разработки оконных приложений. Он предлагает высокую производительность и низкоуровневый доступ к системным ресурсам, что особенно важно для разработки требовательных к производительности приложений, таких как игры или графические редакторы.

C#: C# - язык программирования, разработанный Microsoft, часто используется для создания Windows-приложений. Он предлагает простой синтаксис, широкий набор инструментов и интеграцию с платформой .NET, что делает его мощным инструментом для создания разнообразных приложений, включая настольные, веб-приложения и игры.

Java: Java - мощный и популярный язык программирования, который также широко используется для разработки оконных приложений. Он обладает платформонезависимостью и простотой разработки благодаря использованию виртуальной машины Java (JVM). Java позволяет создавать разнообразные приложения, включая бизнес-приложения, игры и мобильные приложения.

Python: Python предлагает множество библиотек и фреймворков, таких как PyQt и Tkinter, для создания оконных приложений. Python отличается простотой и читабельностью синтаксиса, что делает его идеальным языком для начинающих программистов.

Оконные приложения незаменимы в современном мире по нескольким причинам:

Интерактивность и удобство использования: Оконные приложения предоставляют удобный и интуитивно понятный пользовательский интерфейс, позволяя пользователям взаимодействовать с программой с помощью кнопок, полей ввода, меню и других элементов управления. Это делает приложения более доступными и привлекательными для широкой аудитории.

Мощные возможности: Оконные приложения позволяют разработчикам создавать функционально богатые приложения с расширенными возможностями, такими как многопоточность, обработка событий, визуализация данных и многое другое. Они позволяют создавать сложные интерфейсы, включая графические элементы, анимации, диаграммы и графики, что способствует более эффективной коммуникации с пользователями и улучшает пользовательский опыт.

Оконные приложения представляют собой программы, которые работают в оконном интерфейсе, позволяя пользователю взаимодействовать с приложением через графический интерфейс с использованием окон, кнопок, меню и других элементов. Вот некоторые из популярных видов оконных приложений:

Десктопные приложения:

Десктопные приложения работают на компьютере пользователя и обладают полным доступом к ресурсам операционной системы. Они разрабатываются для конкретной операционной системы, такой как Windows, macOS или Linux, и могут использовать преимущества, предоставляемые этой операционной системой. Примеры десктопных приложений включают текстовые редакторы, графические редакторы, мультимедийные плееры и игры.

Мобильные приложения:

Мобильные приложения предназначены для использования на мобильных устройствах, таких как смартфоны и планшеты. Они разрабатываются для конкретных операционных систем, таких как iOS или Android, и обычно распространяются через соответствующие магазины приложений. Мобильные приложения широко используются в различных областях, включая социальные сети, электронную коммерцию, мультимедиа и здравоохранение.

Веб-приложения:

Веб-приложения работают в браузере и основаны на веб-технологиях, таких как HTML, CSS и JavaScript. Пользователи получают доступ к веб-приложениям через интернет, без необходимости установки дополнительного программного обеспечения. Веб-приложения могут быть доступными на различных платформах и устройствах, так как они не привязаны к конкретной операционной системе. Примеры веб-приложений включают онлайн-офисные пакеты, электронные почтовые клиенты и системы управления контентом.

Кроссплатформенные приложения разрабатываются таким образом, чтобы работать на разных операционных системах без необходимости переписывать исходный код для каждой платформы. Это достигается с использованием кроссплатформенных фреймворков и инструментов разработки, которые обеспечивают возможность создания одного набора кода, который может быть запущен на разных платформах. Примеры кроссплатформенных фреймворков включают React Native, Flutter, Xamarin и Electron.

Игровые приложения:

Игровые приложения являются специальным типом оконных приложений, разработанных для игр. Они могут быть разработаны для десктопа, мобильных устройств или веб-платформ. Игровые приложения используют графику, звуковые эффекты и интерактивность для создания игрового опыта. Разработка игровых приложений требует специализированных инструментов и фреймворков, таких как Unity или Unreal Engine.

Интерактивные информационные системы:

Интерактивные информационные системы представляют собой оконные приложения, которые предоставляют информацию и позволяют пользователю взаимодействовать с ней. Это могут быть приложения для просмотра и редактирования данных, системы управления базами данных, системы отслеживания заказов и т. д. Они часто используются в бизнес-среде для управления данными и автоматизации бизнес-процессов.

Утилиты и инструменты:

Оконные приложения также используются для создания утилит и инструментов, которые помогают пользователям в выполнении конкретных задач. Это могут быть текстовые редакторы, графические редакторы, среды разработки, утилиты для архивации файлов, инструменты для обработки изображений и многое другое.

В зависимости от потребностей и целей разработки, оконные приложения могут иметь различные функциональные возможности, дизайн и сценарии использования. Выбор конкретного вида оконного приложения зависит от платформы, на которой оно будет работать, требований пользователя и целей разработки.

Платформенная независимость: Оконные приложения разработанные с использованием кросс-платформенных языков и фреймворков позволяют запускать приложения на различных операционных системах, таких как Windows, macOS и Linux. Это обеспечивает широкий охват аудитории и повышает доступность приложений для пользователей на разных платформах.

Расширяемость и модульность: Оконные приложения могут быть разделены на модули и компоненты, что облегчает их разработку и поддержку. Модульная структура позволяет разработчикам создавать переиспользуемый код, что повышает эффективность разработки и облегчает добавление новых функций и обновлений в приложение.

Интеграция с другими системами: Оконные приложения могут интегрироваться с другими системами и сервисами, что позволяет обмениваться данными и использовать различные возможности, предоставляемые внешними сервисами. Например, оконные приложения могут взаимодействовать с базами данных, веб-сервисами, сетевыми устройствами и другими внешними ресурсами.

Начинать обучение программированию с разработки оконных приложений имеет несколько преимуществ:

Визуальное представление результата: Разработка оконных приложений позволяет увидеть непосредственные результаты своей работы. Вы можете создать простой интерфейс с кнопками, полями ввода и другими элементами управления и немедленно видеть, как ваше приложение взаимодействует с пользователем.

Мотивация и интерес: Создание оконных приложений может быть захватывающим и интересным процессом. Вы сможете наблюдать, как ваше приложение оживает и становится функциональным. Это может быть мощным стимулом для продолжения обучения и развития в программировании.

Понятный контекст: Разработка оконных приложений часто включает взаимодействие с конкретными задачами и сценариями, которые многим знакомы. Например, создание приложения для управления списком задач или простого текстового редактора. Такие задачи ближе к повседневной жизни и могут быть легче понятны начинающим программистам.

Доступность ресурсов и сообщества: Разработка оконных приложений находится на пике популярности, и для нее существует множество обучающих ресурсов, онлайн-курсов, учебников и форумов. Вы сможете легко найти поддержку и помощь в случае возникновения вопросов или проблем.

Возможность применения в разных областях: Оконные приложения находят применение в различных областях, от бизнес-сферы и разработки игр до научных исследований и образования. Знание разработки оконных приложений открывает возможности для работы в разных отраслях и может быть полезным дополнением к навыкам программирования.

В целом, начинать обучение программированию с разработки оконных приложений может быть интересным и практичным шагом, позволяющим получить навыки, которые могут быть применены в разных сферах и проектах.

* 1. **Основные языки программирования и их свойства, необходимые для разработки оконных приложений**

Интернет – это одна из наиболее значимых изобретений XX века, которая сильно изменила наш мир и повлияла на многие аспекты жизни людей. Изначально он был создан для связи ученых и военных в США в 1960-х годах, но со временем стал доступен широкой публике.

Одним из главных плюсов интернета является возможность получения информации по любому вопросу в любое время и в любом месте. Это способствует повышению уровня образования, улучшению качества жизни и повышению культурного уровня общества в целом.

Также интернет позволяет легко общаться и обмениваться информацией с людьми со всего мира, что помогает расширять кругозор, находить новых друзей и деловых партнеров. Кроме того, интернет предоставляет широкие возможности для развлечений и отдыха – здесь можно найти множество игр, фильмов, музыки и других развлечений.

Однако интернет также имеет свои негативные стороны. Он может приводить к зависимости и утрате реального общения, а также к распространению вредоносной информации, включая фейки и недостоверные данные. Кроме того, интернет может быть использован для противозаконной деятельности, такой как киберпреступность и терроризм.

Интернет постоянно развивается, и новые технологии и приложения становятся доступными для пользователей. Это позволяет совершенствовать существующие сервисы и создавать новые, что положительно сказывается на жизни людей и на развитии общества в целом. Однако постоянное развитие интернета также требует от людей осторожности и внимательности, чтобы избегать потенциальных рисков и негативных последствий.

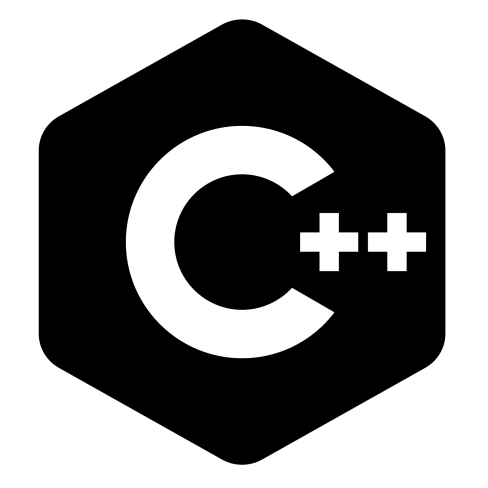
Интернет оказывает огромное влияние на жизнь людей. Он стал неотъемлемой частью современного общества и предоставляет доступ к огромному количеству информации, услуг и возможностей. Он помогает людям учиться, работать, общаться, развлекаться и многое другое.

Интернет был изобретен в 1960-х годах, но первые его прототипы появились еще раньше. Он стал доступен широкой публике в 1990-х годах и с тех пор стал неотъемлемой частью жизни людей. Сначала он был простым средством обмена сообщениями и передачи данных, но в настоящее время он предоставляет доступ к огромному количеству информации, а также предоставляет возможность для общения, работы, развлечений и т.д.

Интернет постоянно развивается и улучшается. Каждый год появляются новые технологии, сервисы и приложения, которые делают его еще более удобным и полезным для людей. Например, сейчас существуют различные сервисы онлайн-образования, онлайн-банкинга, онлайн-покупок и т.д. Также появились новые технологии, такие как искусственный интеллект, интернет вещей и блокчейн, которые могут изменить нашу жизнь еще более сильно в будущем.

Также немаловажную роль сыграли языки программирования, с помощью которых можно создавать любые алгоритмы, программы, сайты, сервисы, приложения и многое другое. Далее будет представлен список и объяснение наиболее популярных современных языков программирования:

C++ - это компилируемый язык программирования, который был создан в 1983 году Бьёрном Страуструпом в AT&T Bell Laboratories. Язык C++ был разработан как расширение языка C, чтобы обеспечить возможности объектно-ориентированного программирования и другие усовершенствования.



C++ стал очень популярным в индустрии программного обеспечения и используется во многих областях, включая разработку операционных систем, приложений, игр, устройств встраиваемых систем, научных и инженерных расчетов и многих других областях.

C++ является мощным языком программирования, который поддерживает множество различных структур данных, включая массивы, списки, стеки, очереди, деревья и многие другие. Также C++ поддерживает различные конструкции языка, такие как условные операторы, циклы, функции, классы и объекты.

C++ имеет множество встроенных структур данных, таких как массивы, списки, очереди и стеки, а также много других, доступных в стандартной библиотеке языка. Он также поддерживает множество различных конструкций, включая условные операторы, циклы, функции, классы и объекты, наследование, полиморфизм и т.д.

Язык C++ широко используется в различных областях, таких как игровая индустрия, научные исследования, финансы, мобильные приложения и многое другое. Следующие известные проекты были написаны на C++:

Операционная система Windows - многие компоненты операционной системы Windows были написаны на C++ для обеспечения высокой производительности и управления ресурсами.

Браузеры Google Chrome и Mozilla Firefox - оба браузера были написаны на C++ для обеспечения быстрой скорости работы и множества возможностей.

Adobe Photoshop - один из самых известных графических редакторов, написанный на C++.

Microsoft Office - набор приложений для работы с текстом, таблицами, презентациями и другими документами, большинство из которых написаны на C++.

MySQL - открытая реляционная база данных, написанная на C++, используется многими веб-сайтами и приложениями.

Qt - кроссплатформенный фреймворк для разработки графических приложений, написанный на C++, широко используется для создания приложений для настольных ПК и мобильных устройств.

Unreal Engine - один из наиболее популярных игровых движков, написанный на C++, широко используется для разработки компьютерных игр.

MATLAB - высокоуровневый язык программирования и интерактивная среда для научных вычислений, написанные на C++, используется для анализа данных, моделирования и симуляций.

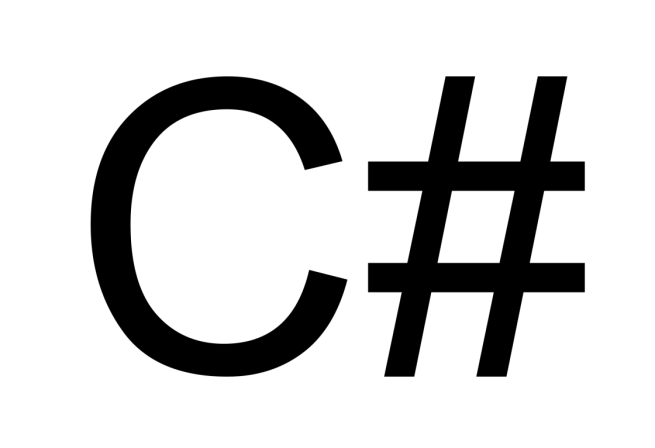
Blender - свободный 3D-графический редактор, написанный на C++, используется для создания анимации, спецэффектов, игр и многого другого.

Amazon Web Services - крупнейшая облачная платформа, написанная на C++, обеспечивает инфраструктуру облачных вычислений и хранения данных для многих компаний.

В последние годы C++ продолжает развиваться, включая новые функции, такие как поддержка многопоточности, улучшения в обработке ошибок и новые библиотеки. C++ также остается одним из самых популярных языков программирования в индустрии, и его использование остается важным во многих областях разработки ПО.

C# (C sharp) - это объектно-ориентированный язык программирования, который был разработан в 2000 году компанией Microsoft для создания приложений для платформы Microsoft .NET Framework. Язык был создан командой разработчиков, которые также работали над языками C++ и Java, и был разработан как язык, сочетающий простоту использования, мощность и скорость выполнения.

C# имеет множество функций, которые делают его мощным и удобным для использования. Он имеет строгую типизацию, поддерживает наследование и полиморфизм, позволяет создавать объекты и использовать компоненты .NET Framework, в том числе Windows Forms, ASP.NET и WPF. Кроме того, C# поддерживает управляемый код, который упрощает процесс сборки мусора и повышает безопасность приложений.



История C# началась в конце 1990-х годов, когда Microsoft решила разработать новый язык программирования, который был бы совместим с платформой .NET Framework. Язык был официально выпущен в 2000 году вместе с первой версией .NET Framework. С тех пор язык постоянно развивался, и на текущий момент он находится на версии 10.

Некоторые известные продукты, написанные на C#, включают в себя:

Microsoft Visual Studio - интегрированная среда разработки, которая использует C# как основной язык для создания приложений под платформы Windows, Android и iOS.

Microsoft Office - некоторые компоненты Microsoft Office, включая Excel и PowerPoint, написаны на C#

Unity - платформа для разработки видеоигр, которая использует C# для написания скриптов.

Windows Presentation Foundation (WPF) - технология, используемая для создания графических пользовательских интерфейсов под Windows, которая написана на C#.

.NET Framework - платформа, разработанная компанией Microsoft для разработки и выполнения программного обеспечения, использующая C# как основной язык программирования.

Xamarin - фреймворк для разработки кроссплатформенных приложений для Android, iOS и Windows, написанных на C#.

Microsoft Dynamics CRM - система управления взаимоотношениями с клиентами, которая использует C# для написания расширений и плагинов.

Это лишь некоторые из множества известных продуктов, созданных с использованием C#. С каждым годом число приложений и продуктов, написанных на этом языке, только растет.

C# используется для создания широкого спектра приложений, в том числе для разработки игр, мобильных приложений, веб-приложений, приложений для управления базами данных и многого другого. Он является одним из наиболее популярных языков программирования, и его непрерывное развитие гарантирует, что он будет оставаться актуальным и востребованным в будущем.

Java - это объектно-ориентированный язык программирования, который был разработан в 1995 году компанией Sun Microsystems (сейчас принадлежит компании Oracle). Java является одним из самых популярных языков программирования в мире, используется для написания приложений на сервере, мобильных приложений, игр и веб-приложений.

Java разработан с упором на простоту, надежность, безопасность и портативность. Он имеет множество функций, включая автоматическое управление памятью, многопоточность, динамическую загрузку классов и многие другие.



Java основан на принципах "Write Once, Run Anywhere" (написать один раз, запустить везде), благодаря которым приложения, написанные на Java, могут быть запущены на любой платформе без необходимости изменения исходного кода. Это достигается за счет того, что Java-приложения компилируются в байт-код, который может быть выполнен на любой платформе, имеющей виртуальную машину Java (JVM).

Java имеет богатую библиотеку классов, которая обеспечивает доступ к большому количеству функций и возможностей, таких как работа с сетью, базами данных, графикой и многим другим. Кроме того, существует множество сторонних библиотек и фреймворков, которые расширяют возможности языка.

Java широко используется в различных сферах, включая разработку программного обеспечения, создание мобильных приложений, веб-разработку и многое другое. Некоторые известные проекты, созданные на Java, включают:

- Android - операционная система для мобильных устройств, созданная на базе Java.

- Minecraft - одна из самых популярных видеоигр, написанная на Java.

- Apache Tomcat - сервер приложений Java для веб-разработки.

- Eclipse - интегрированная среда разработки на Java.

- NetBeans - другая интегрированная среда разработки на Java.

- Hadoop - фреймворк для обработки больших объемов данных.

- OpenJDK - свободно распространяемая реализация Java Development Kit.

- IntelliJ IDEA - еще одна интегрированная среда разработки на Java.

- jMonkeyEngine - игровой движок на Java для создания 3D-игр.

- JIRA - система отслеживания ошибок и управления проектами.

Это лишь некоторые примеры известных проектов, написанных на Java, которые продемонстрировали мощь и гибкость языка программирования.

JavaScript (JS) - это объектно-ориентированный язык программирования, который используется в основном для создания динамических веб-сайтов и интерактивных приложений. Он был создан Бренданом Айком в 1995 году в компании Netscape, как язык сценариев для браузера Netscape Navigator.

JavaScript позволяет создавать интерактивные элементы на веб-страницах, включая анимацию, эффекты, изменение содержимого страницы без перезагрузки и многое другое. Он работает в браузере пользователя, что означает, что код выполняется непосредственно на компьютере пользователя, а не на сервере.



JavaScript имеет несколько структур данных, таких как строки, числа, логические значения и массивы. Он также поддерживает объектно-ориентированное программирование, функциональное программирование, асинхронное программирование и другие методы программирования.

JavaScript широко используется для разработки веб-приложений и сайтов. Некоторые известные проекты, написанные на JavaScript, включают:

Facebook - использует React и React Native, написанные на JavaScript, для создания интерфейсов пользовательского интерфейса.

Twitter - использует фреймворк Bootstrap, написанный на JavaScript, для создания макетов страниц и компонентов пользовательского интерфейса.

Slack - использует фреймворк Electron, написанный на JavaScript, для создания многоплатформенных настольных приложений.

Uber - использует фреймворк Node.js, написанный на JavaScript, для создания масштабируемых серверных приложений и веб-сервисов.

LinkedIn - использует фреймворк AngularJS, написанный на JavaScript, для создания динамических веб-приложений.

Кроме того, существуют множество известных проектов и библиотек, написанных на JavaScript, таких как jQuery, React Native, Vue.js, Node.js, и многие другие.

Далее будет рассмотрен язык программирования на котором будет написан детский фоторедактор.

Python - это высокоуровневый язык программирования общего назначения, который был создан в конце 1980-х годов Гвидо ван Россумом. Язык был назван в честь любимого телешоу Гвидо - "Monty Python's Flying Circus". Python изначально был разработан как язык с простым и понятным синтаксисом, который был бы доступен для широкой аудитории, включая начинающих программистов.

Одним из ключевых преимуществ Python является его читаемость, что делает его привлекательным для разработчиков с любым уровнем опыта. Синтаксис Python напоминает естественный язык, что упрощает понимание кода и его написание.

Python - это интерпретируемый язык программирования, что означает, что он не компилируется в машинный код, а выполняется в интерпретаторе Python. Это также означает, что Python является кроссплатформенным языком и может работать на любой операционной системе.



Python имеет обширную стандартную библиотеку, которая содержит множество модулей для различных задач, таких как работа с файлами, сетевое программирование, создание GUI и многие другие. Кроме того, существует большое количество сторонних библиотек и фреймворков, таких как NumPy, Pandas, Django, Flask, которые значительно расширяют возможности языка.

Python поддерживает несколько структур данных, таких как списки, кортежи, множества, словари, а также объектно-ориентированное программирование, функциональное программирование и др. Кроме того, в Python есть возможность создания генераторов, декораторов и анонимных функций, что упрощает и ускоряет процесс написания кода.

Python также является языком с динамической типизацией, что означает, что тип переменной может быть изменен во время выполнения программы. Это делает Python более гибким, но может также привести к некоторым проблемам в больших проектах.

На данном языке программирования было написано невероятно большое количество проектов, например:

Django — это веб-фреймворк, написанный на Python, который используется для создания крупных веб-приложений. Он предлагает множество готовых компонентов, таких как аутентификация, административная панель, обработка форм и многое другое.

Flask — это минималистичный веб-фреймворк, также написанный на Python. Он предоставляет набор инструментов для создания веб-приложений, но с более простым и легковесным подходом, чем Django.

NumPy — это библиотека для работы с массивами и матрицами в Python. Она предоставляет мощный набор инструментов для выполнения математических операций, таких как вычисление среднего значения, стандартного отклонения, корреляции и многое другое.

SciPy — это библиотека для научных вычислений на Python. Она предоставляет набор инструментов для решения различных задач, таких как оптимизация, регрессия, интерполяция, обработка изображений и многое другое.

Pandas - это библиотека для анализа данных на Python. Она предоставляет удобные инструменты для работы с данными в формате таблицы, такие как фильтрация, сортировка, группировка, объединение и многое другое.

TensorFlow - это библиотека для машинного обучения, разработанная компанией Google на Python. Она предоставляет инструменты для создания и обучения нейронных сетей, а также для выполнения прогнозов и классификации данных.

Pygame - это библиотека для разработки игр на Python. Она предоставляет набор инструментов для создания 2D игр, включая работу с графикой, звуком, вводом и многое другое.

YouTube - веб-сайт видеохостинга YouTube был создан на языке Python, используя фреймворк Django.

Instagram - социальная сеть Instagram была создана на языке Python, используя фреймворк Django.

Python является одним из наиболее популярных и используемых языков программирования в мире. Он отличается своим простым и понятным синтаксисом, что делает его привлекательным для новичков и опытных разработчиков. В этой работе мы рассмотрим основные аспекты синтаксиса Python и расскажем о его ключевых элементах.

Индентация:

В Python блоки кода определяются с помощью индентации, то есть отступов. Это отличает его от других языков программирования, таких как C++ или Java, где используются фигурные скобки для определения блоков кода. Индентация в Python имеет не только визуальное значение, но и синтаксическое. Правильное использование отступов обязательно, чтобы код работал корректно.

Переменные и типы данных:

В Python нет необходимости объявлять типы переменных заранее. Переменные могут быть назначены с помощью оператора «=».Python поддерживает различные типы данных, такие как целые числа, числа с плавающей запятой, строки, списки, кортежи, словари и другие. Операции над переменными выполняются с помощью соответствующих операторов.

Условные выражения и циклы:

В Python есть конструкции для выполнения условных выражений и циклов. Условные выражения выполняются с помощью операторов «if», «elif» и «else». Циклы в Python включают цикл «for», который используется для перебора элементов в последовательности, и цикл «while», который выполняет код до тех пор, пока указанное условие истинно. Функции:

Функции в Python определяются с использованием ключевого слова «def». Они позволяют группировать код для его повторного использования и улучшения структуры программы. Функции также могут принимать аргументы и возвращать значения.

Модули и пакеты:

Python предоставляет возможность организации кода в модули и пакеты. Модуль представляет собой файл с расширением .py, содержащий определения функций, классов и переменных. Пакеты представляют собой директории, содержащие связанные модули.

Обработка исключений:

Python предоставляет механизм обработки исключений, который позволяет программисту предвидеть и обрабатывать возможные ошибки или исключительные ситуации в коде. Используя конструкцию try-except, можно перехватывать и обрабатывать исключения, чтобы предотвратить аварийное завершение программы.

ООП (объектно-ориентированное программирование):

Python поддерживает принципы ООП, что позволяет программистам создавать классы, объекты и использовать наследование, полиморфизм и инкапсуляцию. ООП в Python упрощает разработку сложных программ и способствует повторному использованию кода.

Работа с файлами и операции ввода-вывода:

Python предоставляет мощные возможности для работы с файлами и операциями ввода-вывода. С помощью функций и методов можно открывать, создавать, читать и записывать данные в файлы. Python также поддерживает стандартные потоки ввода-вывода, что позволяет взаимодействовать с пользователем через консоль.

Модули и пакеты:

Python поддерживает модульную структуру программирования, что позволяет разделять код на отдельные модули. Модуль - это файл, содержащий определения функций, классов и переменных. Модули можно импортировать в другие программы для повторного использования кода. Python также поддерживает пакеты, которые объединяют несколько модулей вместе для более организованного иерархического структурирования проектов.

Декораторы:

Python позволяет использовать декораторы - это специальные функции, которые позволяют модифицировать поведение других функций или классов. Декораторы предоставляют удобный способ добавления функциональности к существующему коду без его изменения. Они широко используются для реализации аспектно-ориентированного программирования, кеширования, логирования и других техник.

Многопоточность и асинхронность:

Python предоставляет возможности для работы с многопоточностью и асинхронным программированием. Многопоточность позволяет запускать несколько потоков выполнения параллельно, что полезно для задач, которые могут быть выполнены независимо друг от друга. Асинхронное программирование позволяет создавать эффективные и отзывчивые приложения, способные обрабатывать множество одновременных задач.

Тестирование и отладка:

Python имеет широкий набор инструментов для тестирования и отладки программ. Существуют различные фреймворки для модульного тестирования, функционального тестирования и автоматизации тестирования. Python также предоставляет отладчик, который позволяет искать и исправлять ошибки в программе, облегчая процесс разработки и поддержки кода.

Веб-разработка:

Python активно используется для создания веб-приложений. Существуют различные фреймворки, такие как Django и Flask, которые упрощают разработку веб-приложений, обеспечивая удобные средства для работы с маршрутизацией, шаблонами, базами данных и другими аспектами веб-разработки. Python также поддерживает разработку RESTful API и взаимодействие с различными серверами и клиентами.

Машинное обучение и анализ данных:

Python является одним из наиболее популярных языков программирования в области машинного обучения и анализа данных. Существуют мощные библиотеки, такие как NumPy, Pandas, Matplotlib и Scikit-learn, которые обеспечивают широкие возможности для работы с данными, визуализации, статистического анализа и построения моделей машинного обучения. Python также поддерживает библиотеки глубокого обучения, такие как TensorFlow и PyTorch, которые используются для разработки и обучения нейронных сетей.

Игровая разработка:

Python также используется в игровой разработке. Существуют фреймворки и библиотеки, такие как Pygame и Panda3D, которые позволяют создавать 2D и 3D игры с использованием Python. Благодаря простому и выразительному синтаксису Python становится привлекательным выбором для начинающих разработчиков игр

Разработка мобильных приложений:

Python также может быть использован для разработки мобильных приложений. Существуют фреймворки, такие как Kivy и BeeWare, которые позволяют создавать кросс-платформенные мобильные приложения с использованием Python. Это открывает дополнительные возможности для разработчиков, желающих создавать мобильные приложения с использованием знакомого языка программирования.

Автоматизация задач:

Python предоставляет мощные инструменты для автоматизации задач. Благодаря своей простоте и множеству сторонних библиотек, Python может быть использован для создания скриптов и программ, которые автоматизируют рутинные задачи, обработку данных, управление файлами, взаимодействие с операционной системой и другие задачи, упрощая жизнь разработчиков и пользователей.

Библиотеки для разных областей:

Python имеет огромное сообщество разработчиков, которое активно создает и поддерживает библиотеки для разных областей. Например, для обработки изображений существуют библиотеки, такие как Pillow и OpenCV; для работ с геоданными - библиотеки, такие как GeoPy и Shapely; для разработки интерфейсов - библиотеки, такие как PyQt5 и PySide6; для научных вычислений - библиотеки, такие как NumPy и SciPy; для обработки текстов - библиотеки, такие как NLTK и spaCy; для веб-скрапинга - библиотеки, такие как BeautifulSoup и Scrapy, и так далее. Это позволяет разработчикам использовать готовые инструменты и решения для разных задач, экономя время и упрощая разработку.

Обширное сообщество:

Одним из ключевых преимуществ Python является его обширное и активное сообщество разработчиков. В Python существует множество онлайн-форумов, сообществ, блогов, репозиториев и конференций, где можно получить помощь, задать вопросы, делиться опытом и участвовать в разработке проектов с открытым исходным кодом. Это обеспечивает поддержку и ресурсы для разработчиков на любом уровне, а также способствует постоянному развитию и совершенствованию языка.

Переносимость и доступность:

Python является переносимым языком программирования, что означает, что код, написанный на Python, может быть запущен на различных платформах, таких как Windows, macOS, Linux и других. Более того, Python доступен бесплатно и имеет открытый исходный код, что делает его доступным для широкого круга разработчиков. Python также имеет обширную документацию, богатые ресурсы для самообучения и активное сообщество, что делает его легким и привлекательным языком для изучения и использования.

Легкость обучения:

Синтаксис Python очень читаемый и интуитивно понятный, что делает его легко изучаемым даже для начинающих программистов. Отсутствие сложных символов и структур данных, таких как фигурные скобки или точки с запятой, упрощает процесс понимания и написания кода. Python также предлагает обширную документацию, учебники и онлайн-курсы, что делает его доступным для самообучения.

Интеграция с другими языками:

Python обладает прекрасной интеграцией с другими языками программирования, что позволяет разработчикам использовать его в сочетании с другими языками для создания мощных и гибких систем. Например, с помощью библиотеки ctypes можно вызывать функции на C из Python, а с помощью библиотеки PyJNIus можно работать с кодом Java. Это расширяет возможности языка и позволяет использовать его в разных сценариях разработки.

Большое сообщество библиотек:

Python обладает обширной коллекцией библиотек, которые покрывают практически все области разработки, начиная от научных вычислений и машинного обучения до веб-разработки и игровой индустрии. Например, библиотеки NumPy, Pandas и Matplotlib широко используются в области анализа данных, а Django и Flask - веб-фреймворки, позволяющие разрабатывать мощные веб-приложения. Большое сообщество разработчиков создает и поддерживает эти библиотеки, что упрощает разработку и расширяет возможности Python.

Масштабируемость и производительность:

Python предлагает различные инструменты и подходы для оптимизации производительности кода. Например, с использованием библиотеки Cython можно создавать оптимизированные расширения на языке C для Python, что позволяет улучшить производительность. Также существуют инструменты для параллельного и распределенного выполнения задач, такие как библиотека multiprocessing. Это делает Python подходящим языком для разработки как маленьких скриптов, так и крупных систем.

Активное развитие и поддержка:

Python активно развивается и расширяет свои возможности. Ежегодно выпускаются новые версии языка с улучшениями и новыми функциями. Разработчики Python также обеспечивают поддержку и исправление ошибок, что гарантирует стабильность и надежность языка.

Распространение и популярность:

Python является одним из самых популярных языков программирования в мире. Он используется в различных сферах, от научных исследований и разработки до веб-разработки и искусственного интеллекта. Большое количество компаний и организаций используют Python в своих проектах, что создает большой спрос на разработчиков, знакомых с этим языком.

Большое сообщество и поддержка:

Python имеет активное и дружелюбное сообщество разработчиков. Существуют многочисленные форумы, сообщества, блоги и ресурсы, где можно получить помощь, задать вопросы и обменяться опытом. Также существуют многочисленные онлайн-курсы и обучающие материалы, которые помогают новичкам быстро освоить язык.

Универсальность и мультиплатформенность:

Python является универсальным языком программирования, который может быть использован на различных платформах, включая Windows, macOS и Linux. Код, написанный на Python, может быть перенесен с одной платформы на другую без значительных изменений. Это позволяет разработчикам создавать кросс-платформенные приложения и использовать язык на различных устройствах.

Хорошая документация и учебные ресурсы:

Python предлагает обширную и качественную документацию, которая содержит подробные объяснения и примеры использования различных функций и библиотек языка. Кроме того, существует множество учебников, книг и онлайн-курсов, которые помогают новичкам освоить Python и продвинутым разработчикам расширить свои знания.

Далее будет рассмотрена стандартная библиотека языка программирования Python:

Стандартная библиотека Python является одним из наиболее ценных ресурсов для разработчиков на этом языке. Она предоставляет широкий набор модулей и инструментов, которые значительно упрощают разработку приложений, расширяют функциональность языка и обеспечивают возможность работы с различными аспектами программирования. В данной статье мы рассмотрим некоторые из наиболее популярных модулей стандартной библиотеки Python и их возможности.

Модуль os:

Модуль os предоставляет функции для взаимодействия с операционной системой. Он позволяет осуществлять манипуляции с файлами и директориями, выполнять команды в командной строке, управлять процессами и многое другое. С его помощью можно создавать, перемещать и удалять файлы, проверять существование файлов и директорий, а также управлять переменными окружения.

Модуль datetime:

Модуль datetime предоставляет классы и функции для работы с датами и временем. Он позволяет выполнять различные операции с датами, такие как получение текущей даты и времени, вычисление разницы между датами, форматирование даты и времени в нужном виде и многое другое. Этот модуль особенно полезен при разработке приложений, связанных с обработкой временных данных.

Модуль re:

Модуль re предоставляет функции для работы с регулярными выражениями. Он позволяет выполнять поиск, сопоставление и замену текста на основе заданных шаблонов. С помощью этого модуля можно реализовывать сложные операции по обработке и анализу текста, такие как валидация ввода, поиск ключевых слов, извлечение информации из текстового файла и т.д.

Модуль json:

Модуль json предоставляет функции для работы с данными в формате JSON. Он позволяет сериализовывать объекты Python в JSON-строку и обратно, что является важной возможностью при работе с веб-сервисами, базами данных и другими источниками данных, использующими формат JSON. Этот модуль также поддерживает работу с файлами JSON, чтение и запись данных.

JSON (JavaScript Object Notation) — это легковесный формат обмена данными, основанный на подмножестве языка JavaScript. Он часто используется для передачи данных между клиентом и сервером, хранения настроек, а также во многих других сферах программирования.

JSON представляет собой текстовый формат, который легко читается как человеком, так и компьютером. Он основан на двух структурах данных: объектах и массивах.

Поддержка различных языков программирования: Парсеры JSON доступны для большинства популярных языков программирования, что позволяет использовать JSON в различных средах.

Модуль random:

Модуль random предоставляет функции для работы с случайными числами. Он позволяет генерировать случайные числа, выбирать случайные элементы из списка, перемешивать элементы, моделировать случайные события и многое другое. Этот модуль широко используется в различных областях, таких как игры, моделирование, статистика и тестирование.

Модуль math:

Модуль math предоставляет функции и константы для выполнения математических операций. Он включает в себя функции для вычисления тригонометрических функций, логарифмов, степеней, квадратных корней, а также константы, такие как число π и экспонента. Этот модуль полезен при выполнении сложных математических вычислений.

Модуль csv:

Модуль csv предоставляет функции для работы с данными в формате CSV (Comma-Separated Values). Он позволяет чтение данных из CSV-файлов, запись данных в CSV-файлы и манипуляцию с данными в формате таблицы. Этот модуль часто используется при работе с табличными данными, такими как базы данных или данные из Excel.

Модуль sqlite3:

Модуль sqlite3 предоставляет возможности для работы с базами данных SQLite. Он позволяет создавать, подключаться к базам данных, выполнять SQL-запросы, извлекать и обновлять данные. SQLite является легкой и встраиваемой базой данных, что делает этот модуль удобным инструментом для разработки приложений, требующих хранения и обработки данных.

На основе выше приведённых фактов можно сделать следующий вывод. Стандартная библиотека Python представляет собой мощный набор инструментов, который значительно упрощает разработку приложений на этом языке. Описанные модули являются лишь небольшой частью доступных возможностей стандартной библиотеки. Они обладают широкими функциональными возможностями и предоставляют разработчикам готовые инструменты для работы с файлами, сетью, датами, математическими вычислениями, базами данных и другими аспектами программирования. Использование стандартной библиотеки Python позволяет сократить время разработки, увеличить производительность и надежность приложений.

1. **Разработка оконного приложения фоторедактора**
   1. **Этапы жизненного цикла разработки программного обеспечения на примере оконного приложения**

Фоторедакторы — это программы, предназначенные для редактирования и улучшения фотографий. Они позволяют пользователям изменять яркость, контрастность, цвета, применять эффекты, обрезать изображения, удалять нежелательные элементы и многое другое. Фоторедакторы широко используются как профессиональными фотографами, так и обычными пользователями для создания и улучшения своих фотографий.

Фоторедакторы могут быть различными по своим возможностям и сложности. Они могут варьироваться от простых онлайн-инструментов до мощных программ для настольных компьютеров. Фоторедакторы могут иметь разные функции, такие как ретушь, коррекция цвета, добавление текста, фильтры, коллажи и многое другое.

Фоторедакторы могут быть построены на разных языках программирования и использовать различные библиотеки для работы с изображениями. Некоторые из популярных фоторедакторов включают:

Adobe Photoshop: Один из самых известных фоторедакторов, который предоставляет широкий спектр функций и возможностей для редактирования и обработки фотографий. Он использует свой собственный формат PSD для хранения и обработки изображений. Написан на языке C++ и использует различные технологии и библиотеки для своего функционирования.

Photoshop использует множество библиотек и технологий для реализации своих функций. Некоторые из них включают:

Adobe SDK (Software Development Kit): Adobe предоставляет SDK, который содержит набор инструментов и библиотек для разработки плагинов и расширений для Photoshop. Он позволяет разработчикам расширять функциональность Photoshop и интегрировать его с другими приложениями.

Adobe Imaging Technology: Photoshop использует собственные технологии обработки изображений, которые позволяют выполнять сложные операции редактирования, манипуляции с цветом, ретушь и другие функции. Эти технологии разработаны Adobe специально для Photoshop.

Qt Framework: Photoshop использует Qt, популярный фреймворк для разработки графических приложений. Qt обеспечивает мощные средства для создания пользовательского интерфейса и взаимодействия с пользователем.

OpenGL: Photoshop использует OpenGL для обеспечения аппаратного ускорения и отображения графических элементов интерфейса, таких как окна, кнопки, панели инструментов и другие элементы.

GIMP: Бесплатный и открытый исходный код фоторедактор, который предоставляет множество инструментов и функций для редактирования фотографий. Он доступен для различных операционных систем и поддерживает форматы файлов, такие как JPEG, PNG, GIF и другие.

Разработан на основе языка программирования C и использует различные библиотеки и технологии для своей работы.

Основной язык программирования, на котором написан GIMP, это C. C является мощным языком программирования с низким уровнем абстракции, который обеспечивает высокую производительность и возможности низкоуровневого программирования. Благодаря использованию C, GIMP достигает эффективности и оптимизации при обработке изображений и выполнении сложных операций.

GIMP также использует несколько библиотек и технологий для поддержки своих функций. Некоторые из них включают:

GTK+: GIMP использует библиотеку GTK+ для создания графического пользовательского интерфейса. GTK+ предоставляет набор инструментов для создания окон, кнопок, панелей инструментов и других элементов интерфейса.

GEGL (Generic Graphics Library): GEGL является фреймворком обработки изображений, который используется в GIMP для выполнения различных операций редактирования. Он предоставляет мощные инструменты для манипуляции с цветом, ретуширования, наложения эффектов и других операций обработки изображений.

Cairo: GIMP использует Cairo для рендеринга графических элементов интерфейса и рисования на холсте. Cairo обеспечивает высокую производительность и поддержку различных форматов вывода.

babl: babl является библиотекой для преобразования цветовых моделей и пиксельных форматов. Он используется в GIMP для конвертации и обработки изображений с различными цветовыми пространствами.

GIMP разрабатывается сообществом разработчиков и волонтеров, которые постоянно вносят улучшения и добавляют новые функции. Он является мощным и гибким фоторедактором, доступным для различных операционных систем, и предлагает широкий набор инструментов для редактирования и обработки изображений.

Pixlr: Онлайн-фоторедактор, который предлагает множество инструментов для редактирования изображений. Он имеет простой в использовании интерфейс и поддерживает различные форматы файлов.

Разработан на основе различных веб-технологий, включая HTML, CSS и JavaScript. Он использует современные веб-стандарты и библиотеки для создания интерактивного пользовательского интерфейса и обработки изображений.

Основными языками программирования, которые используются для разработки Pixlr, являются:

HTML (HyperText Markup Language): HTML является основным языком разметки, используемым для создания структуры веб-страницы. Он определяет элементы, теги и атрибуты, которые определяют содержимое и внешний вид веб-приложения.

CSS (Cascading Style Sheets): CSS используется для определения стилей и внешнего вида веб-страницы. Он позволяет управлять цветами, шрифтами, расположением элементов и другими аспектами дизайна интерфейса.

JavaScript: JavaScript является языком программирования, который обеспечивает интерактивность и функциональность веб-приложения. В случае Pixlr, JavaScript используется для обработки изображений, реагирования на действия пользователя и обновления интерфейса в режиме реального времени.

Pixlr также использует различные библиотеки и фреймворки, чтобы упростить разработку и расширить его возможности. Некоторые из них включают:

jQuery: jQuery - это быстрая, компактная и мощная библиотека JavaScript, которая упрощает манипуляцию с HTML-элементами, обработку событий и выполнение AJAX-запросов.

Fabric.js: Fabric.js - это библиотека для работы с холстом HTML5, которая обеспечивает возможность создания и манипуляции с объектами на холсте, включая изображения, формы и текст.

WebAssembly: WebAssembly - это двоичный формат исполняемого кода для веб-приложений, который позволяет выполнять высокопроизводительные вычисления в браузере. Pixlr может использовать WebAssembly для ускорения обработки изображений и выполнения сложных операций.

Lightroom: Программа для обработки фотографий, разработанная компанией Adobe. Он предоставляет инструменты для редактирования, организации и обработки фотографий в больших объемах.

Lightroom разработан на основе следующих технологий и языков программирования:

C++: Lightroom написан на языке программирования C++. C++ - это мощный язык программирования, который обеспечивает высокую производительность и эффективное управление памятью. Он широко используется для разработки приложений с высокими требованиями к производительности, таких как графические редакторы.

Lua: Lua - это легкий и гибкий скриптовый язык программирования, который используется в Lightroom для реализации скриптовых возможностей и расширений. Lua позволяет пользователям создавать собственные скрипты для автоматизации задач и расширения функциональности приложения.

Qt: Lightroom использует фреймворк Qt для создания пользовательского интерфейса и обеспечения переносимости на различные операционные системы. Qt предоставляет набор инструментов и библиотек для разработки графических приложений с использованием C++.

Adobe Camera Raw: Lightroom также использует Adobe Camera Raw, который представляет собой плагин для обработки и конвертации RAW-файлов в формате Digital Negative (DNG). Adobe Camera Raw написан на основе различных технологий, включая C++, Adobe Photoshop SDK и другие внутренние технологии Adobe.

Lightroom предлагает богатый набор инструментов для редактирования и управления фотографиями, включая функции коррекции цвета, настройки экспозиции, улучшения деталей и многое другое. Благодаря использованию C++ и других технологий, Lightroom обеспечивает высокую производительность и возможности работы с большими объемами фотографий.

Canva: Веб-приложение для создания дизайна, которое также включает инструменты для редактирования фотографий. Оно предоставляет шаблоны, фильтры и другие инструменты для создания эстетически привлекательных изображений. Lightroom разработан на основе следующих технологий и языков программирования:

C++: Lightroom написан на языке программирования C++. C++ - это мощный язык программирования, который обеспечивает высокую производительность и эффективное управление памятью. Он широко используется для разработки приложений с высокими требованиями к производительности, таких как графические редакторы.

Lua: Lua - это легкий и гибкий скриптовый язык программирования, который используется в Lightroom для реализации скриптовых возможностей и расширений. Lua позволяет пользователям создавать собственные скрипты для автоматизации задач и расширения функциональности приложения.

Qt: Lightroom использует фреймворк Qt для создания пользовательского интерфейса и обеспечения переносимости на различные операционные системы. Qt предоставляет набор инструментов и библиотек для разработки графических приложений с использованием C++.

Adobe Camera Raw: Lightroom также использует Adobe Camera Raw, который представляет собой плагин для обработки и конвертации RAW-файлов в формате Digital Negative (DNG). Adobe Camera Raw написан на основе различных технологий, включая C++, Adobe Photoshop SDK и другие внутренние технологии Adobe.

Lightroom предлагает богатый набор инструментов для редактирования и управления фотографиями, включая функции коррекции цвета, настройки экспозиции, улучшения деталей и многое другое. Благодаря использованию C++ и других технологий, Lightroom обеспечивает высокую производительность и возможности работы с большими объемами фотографий.

Фоторедакторы работают с изображениями, которые представлены в виде растровых или векторных файлов. Растровые изображения состоят из пикселей, каждый из которых содержит информацию о цвете и яркости. Векторные изображения представлены в виде графических объектов, таких как линии, кривые и фигуры, и хранятся в виде математических формул.

Для работы с изображениями фоторедакторы используют различные алгоритмы и операции. Некоторые из основных функций, которые предоставляют фоторедакторы, включают:

Редактирование цвета и тона: Фоторедакторы позволяют пользователю изменять яркость, контрастность, насыщенность и оттенок изображения. Это позволяет улучшить качество фотографии и достичь нужного эффекта.

Ретушь и исправление: С помощью фоторедакторов можно удалять дефекты, такие как пятна, морщины, пыли и шумы на фотографиях. Также можно выполнять ретушь кожи, удалять красные глаза и выполнять другие исправления.

Обрезка и масштабирование: Фоторедакторы позволяют изменять размер изображения и обрезать его для получения нужной композиции. Это полезно при подготовке фотографий для публикации в социальных сетях или печати.

Добавление текста и графики: Фоторедакторы предоставляют возможность добавлять текстовые надписи, подписи и водяные знаки к фотографиям. Также можно создавать коллажи, добавлять рамки и другие графические элементы.

Применение фильтров и эффектов: Фоторедакторы имеют набор предустановленных фильтров и эффектов, которые позволяют изменить внешний вид фотографии. Это может быть эффект размытия, черно-белый стиль, сепия и многие другие.

* 1. **Процесс разработки оконного приложения фоторедактора**

Жизненный цикл программного обеспечения (Software Development Life Cycle, SDLC) представляет собой последовательность этапов и процессов, которые проходит программное обеспечение, начиная от его создания и до его вывода в эксплуатацию и поддержки. Ниже представлены основные этапы жизненного цикла программного обеспечения на примере разработки оконного приложения:

Анализ и планирование:

Определение требований: Заказчик и разработчики определяют функциональные и нефункциональные требования к оконному приложению.

Анализ изначальной концепции: Разработчики проводят анализ концепции приложения, определяют его цели, аудиторию, основные функциональности и ограничения.

Планирование: создается план разработки, определяются ресурсы, график работ и оценка затрат.

Проектирование:

Архитектурное проектирование: Разрабатывается общая архитектура приложения, определяются его компоненты, взаимодействие между ними и структура данных.

Детальное проектирование: Проектируются каждый компонент и модуль приложения, определяются интерфейсы, алгоритмы и структуры данных.

Разработка:

Написание кода: Разработчики пишут и тестируют код приложения, используя выбранный язык программирования и соответствующие инструменты и библиотеки.

Тестирование модулей: Каждый модуль приложения проходит тестирование на соответствие требованиям и корректность работы.

Тестирование и отладка:

Функциональное тестирование: Проверка работы приложения на соответствие функциональным требованиям.

Интеграционное тестирование: Тестирование взаимодействия между компонентами приложения.

Системное тестирование: Тестирование приложения в целом, в различных сценариях использования.

Отладка и исправление ошибок: если обнаружены ошибки, разработчики ищут и исправляют их.

Выпуск и развертывание:

Упаковка приложения: Приложение упаковывается в удобный формат для установки и использования, например, в виде исполняемого файла или инсталляционного пакета.

Развертывание: Приложение устанавливается на целевые устройства или серверы, где пользователи смогут его запустить и использовать.

Эксплуатация и поддержка:

Работа приложения: Пользователи могут использовать оконное приложение в соответствии с его назначением и функциональностью.

Техническая поддержка: Разработчики могут предоставлять техническую поддержку пользователям, отвечать на вопросы, исправлять ошибки и выпускать обновления.

Обновление и сопровождение:

Внесение изменений: По мере необходимости разработчики могут выпускать обновления, добавлять новые функции, исправлять ошибки или улучшать производительность приложения.

Сопровождение: Приложение может требовать регулярного сопровождения, включая обновление зависимостей, исправление уязвимостей и поддержку новых версий операционных систем и другого программного обеспечения.

Фаза вывода из эксплуатации:

Прекращение использования: в некоторых случаях оконное приложение может быть выведено из эксплуатации, например, если его функциональность устарела, существуют более эффективные альтернативы или появились изменения в требованиях пользователей.

Архивирование и сохранение данных: перед полным выводом приложения из эксплуатации необходимо архивировать и сохранить данные, связанные с приложением, чтобы обеспечить сохранность информации и возможность ее восстановления в будущем.

Документация и оценка: важно составить полную документацию о приложении, чтобы предоставить информацию будущим разработчикам или пользовательской службе поддержки. Также может быть проведена оценка приложения для выявления улучшений или определения причин его вывода из эксплуатации.

Каждый этап жизненного цикла программного обеспечения включает в себя ряд действий и процессов, которые помогают разработчикам создать качественное и стабильное оконное приложение. Этот жизненный цикл позволяет планировать, разрабатывать, тестировать, выпускать, эксплуатировать и поддерживать приложение в течение его жизни.

Анализ и планирование:

На этапе практической реализации проекта, было решено следовать каскадной модели разработки программного обеспечения. Применение данного типа модели ЖЦ ПО обусловлено линейностью разработки, относительной простотой проекта, а также отсутствием необходимости согласовывать каждый этап разработки с заказчиком, коллегами.

Первым этапом является этап генерации идеи будущего программного продукта. В основу идеи легла разработка оконного приложения фоторедактора с небольшим, но разнообразным функционалом. Приложение будет рассчитано на детей, обучающихся на фотокурсах или аудиторию, занимающуюся начальной обработкой графических объектов.

Анализ существующих оконных приложений показал, что их интерфейс обладает сложной структурой и не является интуитивно понятным для неопытного пользователя. Примеры таких приложений, как Photoshop, Canva, Ligthroom и других аналогов труднодоступны, имеют ограниченный бесплатный функционал, либо полностью отсутствуют на российском рынке приложений.

Минимальный жизнеспособный продукт (MVP) должен соответствовать следующим требованиям:

Минималистичный интерфейс

Минимальный набор необходимых функций

Простота

Доступность

Кроссплатформенность

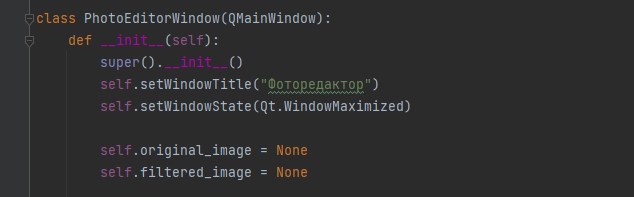
Возможность легко сохранять преобразованное изображение

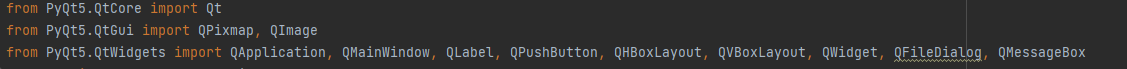
Был выделен следующий набор функций: Выбор папки, Поворот изображения на 90° влево или вправо, Отражение, Эффект ч/б, Изменение резкости, Размытие по Гауссу, Сохранение результата.

Проектирование:

По заявленному функционалу был составлен макет фоторедактора

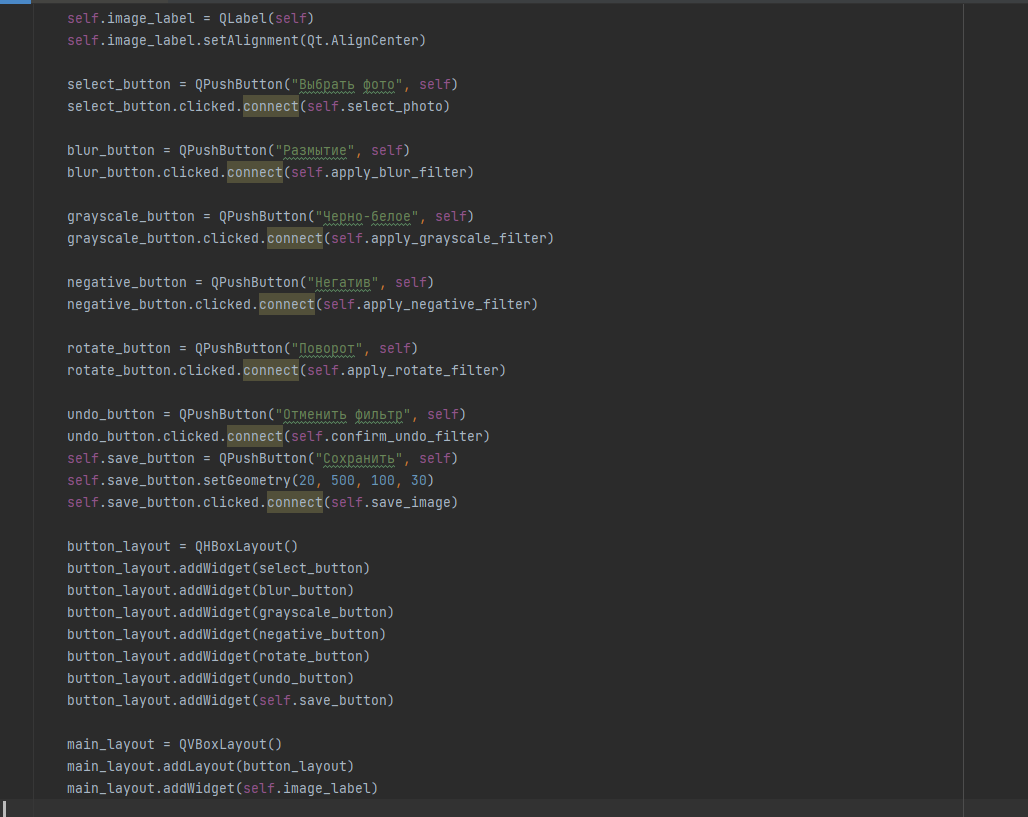
Разработка:

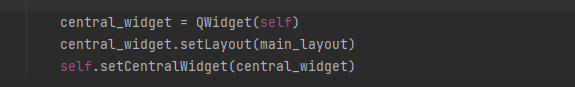
Сам фоторедактор относительно кода был представлен в виде класса с методами редактирования фотографии: 

Для разработки интерфейса была выбрана библиотека PyQt5

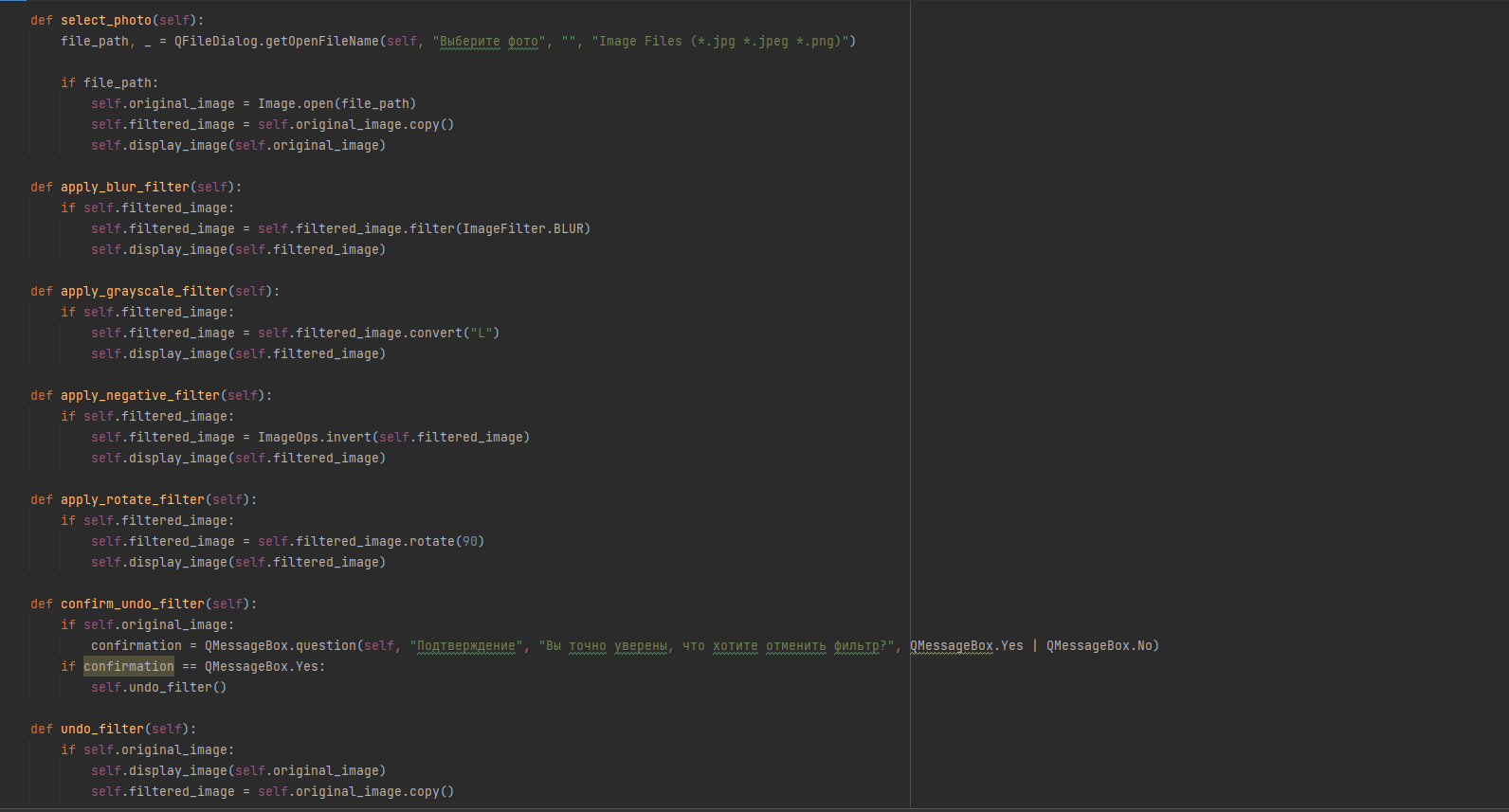
А для редактирования фотографий была выбрана библиотека Pillow

Ниже будет приведён код, для построения окна





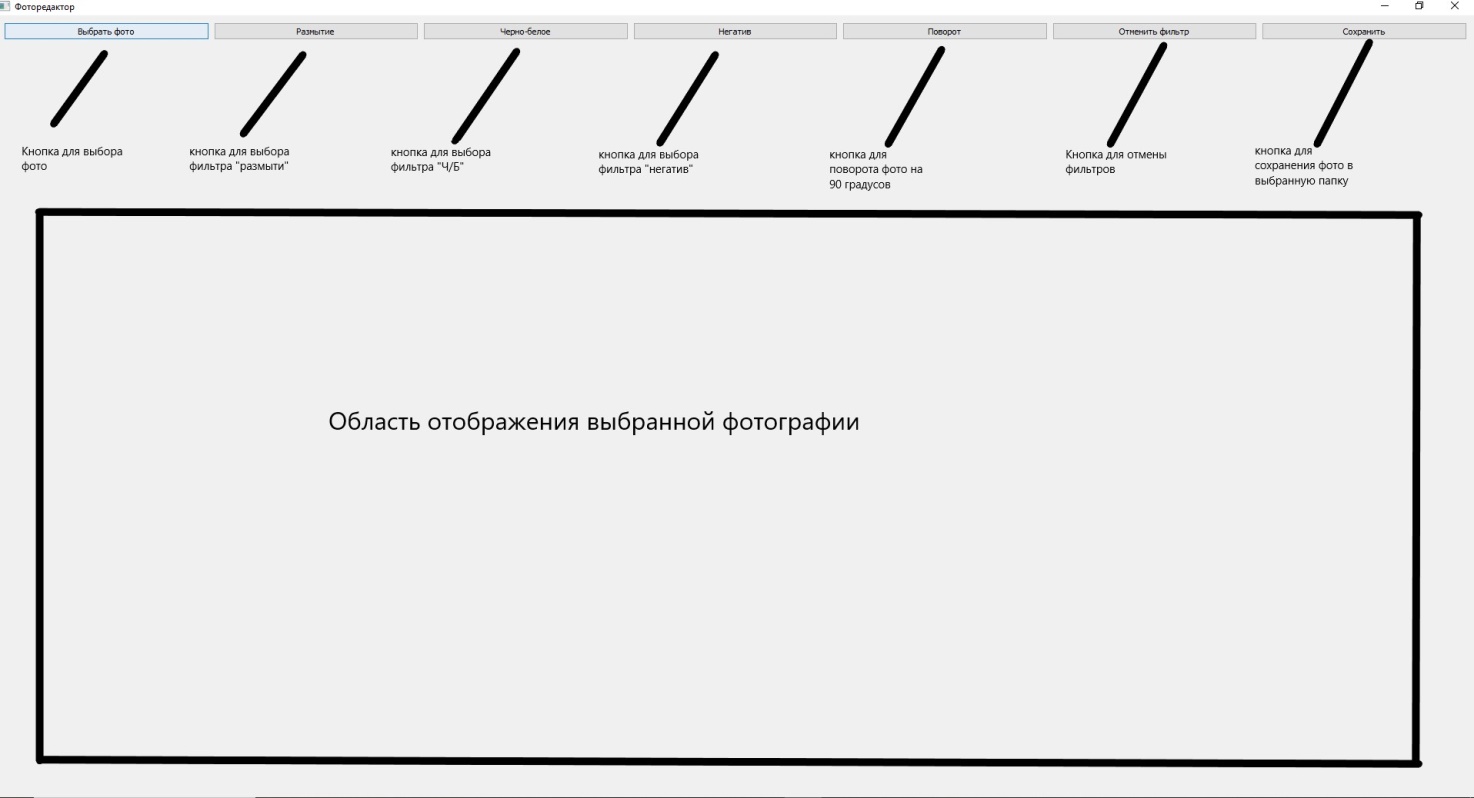
Далее будут представлены фрагменты кода связанные с фильтрами



В результате разработки и компиляции приложения была собрана первая версия фоторедактора.

Тестирование и отладка:

В первой версии приложения были выявлены недочёты: невозможность выбора конкретной фотографии, невозможность сохранить фото в любую папку, и относительно маленькая демонстрация оригинала фотографии

Выпуск и развертывание:

ВСЕ СДЕЛАЛ, ВСЕ РАБОТАЕТ, КАК МОЖНО УЛУЧШИТЬ

**Заключение**

НАПИСАТЬ НА СТРАНИЦУ

**Список использованных источников**

1. Вандер П.Д. Python для сложных задач. Наука о данных и машинное обучение. Руководство/ Плас Джейк Вандер – Изд. Питер, 2023 – 567 с.
2. Лутц М. Изучаем Python, том 1, 5-е изд.: Пер. с англ. — СПб.: ООО «Диалектика», 2019. — 832 с.
3. Мэтиз Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд./Эрик Мэтиз — СПб.: Питер, 2020. — 512 с.

"PyQt5. Создание оконных приложений на Python" Б. М. Харвани

"Изучаем Python. Практическое руководство по программированию" Марк Лутц

"Программирование на Python. Основы компьютерной науки" Джон Зелле

"Быстрое создание GUI-приложений на Python и Qt. Полное руководство по PyQt программированию" Марк Саммерфилд

"Графическое программирование на Python. Создание 2D и 3D изображений" Б. М. Харвани

"PyQt5. Руководство" Ян Боднар

"Разработка GUI-приложений на Python с использованием Tkinter. Книга рецептов" Бхаскар Чаудхари

"wxPython. Книга рецептов" Коди Прекорд и Ноэль Рэппин.