муниципальное бюджетное общеобразовательное учреждение

средняя школа №10

**ПРОЕКТ**

**«Трекер события»**

**(тип проекта - прикладной)**

Выполнил:

Денисов Михаил Романович,

учащийся 10 класса

Руководитель:

Лосева Наталья Александровна,

учитель информатики

г.о.г. Арзамас, 2025 год

**ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ ……………………………………………………………5

ГЛАВА 1. Теоретические аспекты ………………………………………………6

1.1. Основные сведения о языке программирования Python ………………………6

# 1.2. Система программирования PyCharm ……………………………………………………..8

# 1.3. Описание библиотек, используемых в проекте…………………………………………….12

1.4 Программа для разработки графических интерфейсов………………………12

ГЛАВА 2. Практическое применение знаний ……………………………………14

2.1. Изучение python……………………………………………………… 14

2.2. QTdesigner ………………………………………………………14

ЗАКЛЮЧЕНИЕ …………………………………………………………16 СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ…………………………17 ПРИЛОЖЕНИЯ …………………………………………………………………

# ПЛАН ВЫПОЛНЕНИЯ ПРОЕКТА

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Действие | Сроки | Необходимые ресурсы | Ожидаемый результат | Корректировка плана | Отметка о выполнении |
| 1 | Выбор темы | 15.09.24 –  20.09.24 | Компьютер  Интернет | Задумка проекта |  |  |
| 2 | Определение этапов работы | 21.09.24  -  26.09.24 | Word | План работы |  |  |
| 3 | Обоснование актуальности | 27.09.24  -  07.10.24 | Word | Написать актуальность проекта |  |  |
| 4 | Обозначить цель | 07.10.24  -  18.10.24 | Word | Написать цель проекта |  |  |
| 5 | Обозначить задачи | 19.10.24  -  29.10.24 | Word | Написать задачи проекта |  |  |
| 6 | Поиск нужной литературы | 30.10.24  -  01.11.24 | Интернет | Нахождение литературы |  |  |
| 7 | Изучение литературы | 1.11.24  -  16.11.24 | Интернет | Пополнение себя новыми знаниями |  |  |
| 8 | Создание чернового варианта проекта | 17.11.24  -  21.11.24 | Pycharm | Создание пробного варианта проекта |  |  |
| 9 | Исправление ошибок | 27.11.24  -  05.12.24 | Pycharm  Интернет | Доработка программы |  |  |
| 10 | Улучшение дизайна | 05.12.24  -  10.12.24 | Pycgarm  Qtdesigner  Интернет | Доработка программы |  |  |
| 11 | Подготовка к предзащите | 11.12.24  -  24.02.25 | Интернет  Word | Готовность к предзащите |  |  |
| 12 | Предзащита | 25.02.25 | Школа | Защита |  |  |
| 13 | Исправление ошибок | 25.02.25  -  01.04.25 | Word  Pycharm  Интернет | Исправление недочетов |  |  |
| 14 | Защита | 01.04.25  -  26.04.25 | Школа | Защита |  |  |

# ВВЕДЕНИЕ

**Актуальность проекта** по отслеживанию выбранной даты можно обосновать несколькими ключевыми аспектами:

1. Управление временем: в современном мире, где информация и события стремительно меняются, эффективное управление временем становится критически важным. Проект по отслеживанию выбранной даты помогает пользователям не забывать о важных событиях, таких как дни рождения, дедлайны по проектам и другие значимые даты.

2 Личное и профессиональное планирование: Проект может быть полезен как для личного использования, так и для профессиональной деятельности. Он может помочь в планировании встреч, проектов и мероприятий, что особенно актуально для команд, работающих удаленно или в разных часовых поясах.

3. Адаптация к индивидуальным потребностям: Проект может быть адаптирован под различные группы пользователей — от студентов до бизнесменов. Возможность настройки уведомлений и напоминаний под личные предпочтения делает его более привлекательным и полезным.

4. Социальные аспекты: Отслеживание значимых дат может способствовать укреплению социальных связей. Напоминания о днях рождения и других важных событиях помогают поддерживать отношения с друзьями и близкими.

5. Обратная связь и улучшение сервиса: Внедрение функционала для получения обратной связи от пользователей позволяет постоянно улучшать проект, добавляя новые функции и улучшая пользовательский интерфейс.

Таким образом, проект по отслеживанию выбранной даты не только отвечает потребностям современного человека в организации времени, но и способствует повышению качества жизни за счет улучшения управления личными и профессиональными задачами.

Этим приложением могут воспользоваться многие категории людей: студенты, профессионалы в определенной области, предприниматели, родители и т.д.

**Целью проекта** выступает создание трекера событий с помощью языка программирования Python.

**Задачи:**

* изучить язык программирования Python;
* прочитать документацию о модулях Python
* выбрать нужные для моего проекта
* подробнее изучить их
* разработать визуал
* написать пробный вариант программы
* исправить появившиеся ошибки и баги
* довести программу до рабочего состояния

# Глава 1.Теоретические аспекты создания визуальной новеллы

# Основные сведения о языке программирования Python

Программирование - это процесс создания [компьютерных программ](https://bigenc.ru/c/komp-iuternaia-programma-c84ead) или [программного обеспечения](https://bigenc.ru/c/programmnoe-obespechenie-ac0e55), а также дисциплина, изучающая методы и приёмы создания и развития компьютерных программ (более точное название дисциплины – инженерия программного обеспечения или программная инженерия).(1)

Для написания программ используются различные языки программирования. Вот самые востребованные языки в современном мире:

1. Python
2. [JavaScript](https://blog.skillfactory.ru/samye-populyarnye-yazyki-programmirovaniya-2024/#javascript)
3. [Java](https://blog.skillfactory.ru/samye-populyarnye-yazyki-programmirovaniya-2024/#java)
4. [PHP](https://blog.skillfactory.ru/samye-populyarnye-yazyki-programmirovaniya-2024/#php)
5. [Kotlin](https://blog.skillfactory.ru/samye-populyarnye-yazyki-programmirovaniya-2024/#kotlin)
6. [TypeScript](https://blog.skillfactory.ru/samye-populyarnye-yazyki-programmirovaniya-2024/#typescript)
7. C и C++
8. [C#](https://blog.skillfactory.ru/samye-populyarnye-yazyki-programmirovaniya-2024/#c)
9. [Go](https://blog.skillfactory.ru/samye-populyarnye-yazyki-programmirovaniya-2024/#go)

Самыми популярными являются Python, C подобные и java.

Я программирую на языке python, поэтому хотел бы рассказать о нем более подробно.

История языка программирования Python началась в конце 1980-х. [Гвидо ван Россум](https://ru.wikipedia.org/wiki/%D0%A0%D0%BE%D1%81%D1%81%D1%83%D0%BC,_%D0%93%D0%B2%D0%B8%D0%B4%D0%BE_%D0%B2%D0%B0%D0%BD) задумал [Python](https://ru.wikipedia.org/wiki/Python) в 1980-х годах а приступил к его созданию в декабре 1989 года в центре математики и информатики в Нидерландах. Язык Python был задуман как потомок [языка программирования ABC](https://ru.wikipedia.org/wiki/ABC_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), способный к обработке исключений и взаимодействию с операционной системой [Amoeba](https://ru.wikipedia.org/wiki/Amoeba_(%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)). Ван Россум является основным автором Python и продолжал выполнять центральную роль в принятии решений относительно развития языка вплоть до 12 июля 2018 года.

Версия Python 2.0 была выпущена 16 октября 2000 года и включала в себя много новых крупных функций — таких как полный [сборщик мусора](https://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и поддержка [Unicode](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4). Однако наиболее важным из всех изменений было изменение самого процесса развития языка и переход на более прозрачный процесс его создания.

Первая обратно-несовместимая версия Python 3.0 была выпущена 3 декабря 2008 года после длительного периода тестирования. Многие её функции были портированы и обратно совместимы с Python 2.6 и Python 2.7[2].

Для каких целей используется python?

### **Веб-разработка**

Чаще всего Python используют в веб-разработке. Для него написано множество Фреймворков: FastAPI, Flask, Tornado, Pyramid, TurboGears, CherryPy и, самый популярный, Django.

### **Научные исследования и машинное обучение**

Благодаря лаконичности, простому синтаксису и богатому набору инструментов Python стал любимым языком учёных. Есть специальные библиотеки для Python, которые позволяют строить графики, проводить исследования и вычисления:

### **Тестирование**

Ещё одна область применения Python — автоматизация тестирования. Многие специалисты по автоматизации QA выбирают Python из-за его простоты. Он отлично подходит тем, кто имеет небольшой опыт в разработке приложений. Развитое сообщество, логичный синтаксис и удобочитаемость упрощают процесс обучения.

### **Десктоп-приложения**

Хоть язык не компилируется, с его помощью всё же можно создать десктопные программы.

### **Мобильные приложения**

Мобильная разработка на Python менее популярна. Для Android чаще используют Java, C#, C++ или Kotlin, а для iOS — Swift или Objective-C. На Python обычно программируют серверную часть приложения. Например, клиент «Инстаграма»\* для iOS написан на Objective-C, а сервер — на Python. Тем не менее у Python есть фреймворки для разработки кроссплатформенных мобильных GUI-приложений: Kivy и BeeWare.

### **Игры**

Многие компьютерные игры полностью или частично написаны на Python. Существует заблуждение, что этот язык не подходит для серьёзных проектов, но на самом деле он использовался в разработке таких хитов, как:

* Battlefield 2;
* World of Tanks;
* Civilization IV;
* The Sims 4;
* EVE Online[3].

Чем python отличается от других языков?

1. Читабельность и простота. Python известен своей чистой и лаконичной синтаксической структурой, что делает его легко читаемым и понятным даже для новичков. Этим он выгодно отличается от C++ или Java с более сложным и запутанным синтаксисом.
2. Большая стандартная библиотека. Python имеет обширную стандартную библиотеку, которая включает множество модулей и пакетов для различных задач, таких как веб-разработка, обработка данных, машинное обучение, сетевое программирование и многое другое. Это сокращает время, необходимое для разработки программ, и уменьшает зависимость от сторонних библиотек.
3. Кроссплатформенность. Python является кроссплатформенным языком. Программы на «питоне» можно запускать в различных операционных системах (Windows, macOS и Linux) без изменения исходного кода.
4. Динамическая типизация. В отличие от статически типизированных языков (Java или C++), Python использует динамическую типизацию. Прием позволяет программистам не объявлять тип переменной заранее. В итоге мы получаем более гибкий код и сокращенное количество строк.
5. Интерпретируемость. Python — интерпретируемый язык. Его код выполняется построчно, без необходимости компиляции. Это упрощает процесс разработки и облегчает отладку программ.
6. Большое сообщество. Python обладает одним из самых больших и активных сообществ разработчиков, что способствует созданию множества ресурсов, таких как учебники, курсы, библиотеки и фреймворки, которые помогают ускорить процесс разработки и обучения.
7. Применимость в различных областях. Python широко используется в различных сферах, таких как веб-разработка, научные исследования, анализ данных, машинное обучение, сетевое программирование, автоматизация и многих других. Это делает Python универсальным и гибким языком программирования, подходящим для широкого круга задач и проектов.
8. Поддержка множественных парадигм программирования. Python поддерживает несколько парадигм программирования: объектно-ориентированное, процедурное и функциональное. Это позволяет разработчикам выбирать наиболее подходящий метод для решения конкретной задачи и создания гибких, масштабируемых приложений.
9. Широкое использование в индустрии. Python используется множеством крупных компаний (Google, Facebook, Instagram, Spotify), что говорит о высоком уровне доверия к языку и его применимости для решения сложных задач.
10. Открытый исходный код Python доступен для просмотра и модификации. Это позволяет разработчикам вносить свой вклад в развитие языка и использовать его в коммерческих проектах без оплаты лицензионных сборов[4].

# 1.2. Система программирования PyCharm

Самая удобная программа для программирование – это PyCharm.

**PyCharm** — это интегрированная среда разработки на языке Python, созданная компанией JetBrains. Среди питонистов эта IDE [вторая по популярности](https://skillbox.ru/media/code/komanda-python-i-jetbrains-proveli-ezhegodnyy-opros-pitonistov/?utm_source=media&utm_medium=link&utm_campaign=all_all_media_links_links_articles_all_all_skillbox) после редактора кода Visual Studio Code: как основную её используют 31% разработчиков.

В PyCharm есть все инструменты, чтобы **писать, отлаживать и тестировать код**. Например, можно быстро исправить программу сразу в нескольких местах, а встроенный форматер приведёт её в соответствие со стандартом PEP 8.

Также IDE позволяет использовать в проектах другие языки программирования, синхронизировать код с системами контроля версий и развёртывать его.

Базовый набор функций PyCharm можно расширять с помощью **плагинов**, которые позволяют, например, настраивать внешний вид интерфейса и подключать дополнительные инструменты[9].

**1.** Docker

Docker — это известный инструмент контейнеризации Python, представленный в 2013 году *Соломоном Хайксом.* Этот инструмент используется во время развертывания любого проекта и в основном работает на всех ОС, включая Linux, Mac и Windows.

Docker — идеальный выбор, когда речь идет о доставке, благодаря своей переносимости и эффективности, поскольку он позволяет автоматизировать развертывание внутри контейнера. Разработчики также находят простоту настройки технической конфигурации, которая напрямую ограничивает использование дополнительных ресурсов. Вот несколько причин, по которым он по-прежнему отлично работает для разработчиков Python, а также получил рейтинг 4,6 (по популярности) на крупных веб-сайтах. Ниже также приведен прикрепленный снимок экрана для справки:

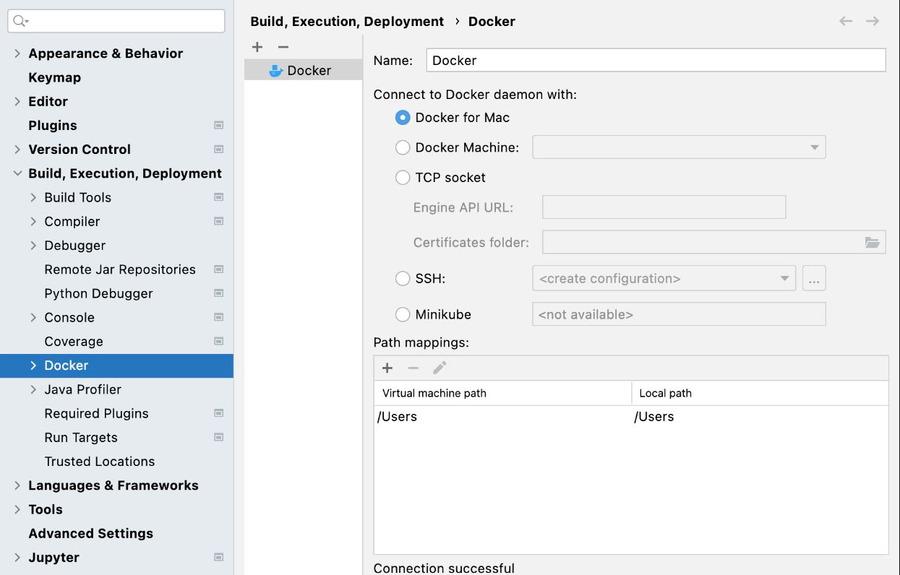


Рис.1. Docker

Однако Docker также предлагает некоторые из наиболее важных функций, которые необходимо обратить внимание ниже:

* Возможность оформления формы
* Меньше нагрузки на оборудование
* Swarm — они работают как связующее звено кластера внутри Docker.

## 2. Markdown

Этот плагин является идеальной комбинацией для редактирования файлов в разметке, а также позволяет пользователям использовать пользовательские таблицы стилей для создания предпросмотров во время выполнения. Уценка также относится к нижеперечисленным функциям:

* Обеспечивает поддержку радиоприемника с кодами, включая изображения или исходный код.
* Предоставляет интерфейс командной строки.
* Он может предоставлять данные о выходных в различных форматах, таких как теги в стиле HTML или XHTML.
* Markdown включает в себя несколько языков, поддерживающих Unicode.
* Этот плагин поддерживает языковые диаграммы (такие как Mermaid или PlantUML)

Для лучшего понимания вы также можете просмотреть прикрепленный снимок экрана:

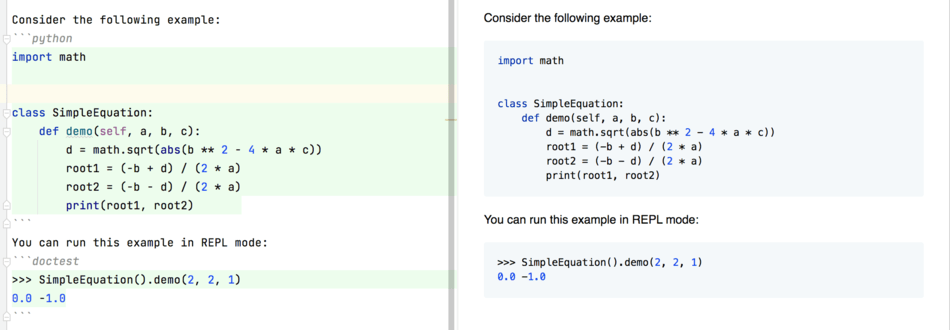


Рис.2. Markdown

## 3. Rainbow Brackets

Те, кто любит кодирование, должны использовать этот инструмент, он предлагает очень яркую особенную сложную скобку, которая сама по себе является чисто вложенными играми. Самое лучшее в этом инструменте то, что его можно использовать в нескольких странах, включая:

* [Ява](https://www.geeksforgeeks.org/java-tutorial/)
* [Скала](https://www.geeksforgeeks.org/scala-tutorial-learn-scala-with-step-by-step-guide/)
* [Котлин](https://www.geeksforgeeks.org/kotlin-programming-language/)
* [Питон](https://www.geeksforgeeks.org/python-programming-language/learn-python-tutorial/)
* [HTML](https://www.geeksforgeeks.org/html-basics/)
* [XML](https://www.geeksforgeeks.org/xml-basics/) и т.д.

Кроме того, он хорошо подходит для Android Studio, IntelliJ IDEA, PhpStorm и т. д. д. Ниже вы найдете приведенный скриншот:



Рис.3. Rainbow Brackets

## 4.  IdeaVim

IdeaVim был представлен *Jetbrains* и представляет собой расширение, используемое разработчиками в качестве эмуляции IntelliJ и других IDE. Это расширение предлагает основные ключевые функции, такие как Mode, Motion, Command, Keymap и т. д. д. Оно предлагает режимы для обычного редактирования, визуальный режим, режим для вставок и т. д. д. Репозиторий можно скачать с GitHub, перейдя по ссылке ниже:

Этот инструмент также предлагает некоторые из самых простых команд, в том числе:

* ct)
* ciw
* #
* ~
* зз

Ниже вы также можете найти скриншот для лучшего ознакомления:

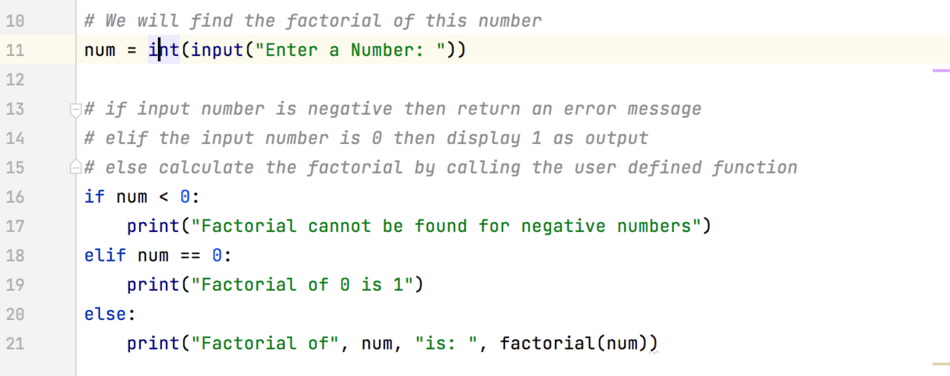


Рис.4. IdeaVim

Помимо этого, IdeaVim обладает и другими функциями:

* Поддержка нескольких языков
* Простая интеграция с GIT
* Удивительные инструменты рефакторинга, а также поддержка фреймворков и библиотек.

## 5. String Manipulation

При работе над определением становится сложнее ввести код построчно, и тут-то и вступает в игру это простое расширение. С помощью String Manipulation вы сможете манипулировать текстами, а также позволяет выполнять сортировку с учетом регистра, то есть Az, zA, AZ, ZA.

Разработчики Python активно используют этот инструмент, и он действительно экономит много времени, так как им не нужно удалять нежелательные тексты или пробелы из кода строки. Также ниже вы найдете прикрепленный скриншот для справки:

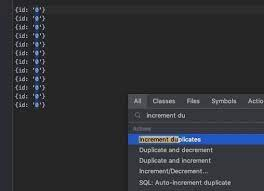


Рис.5. String Manipulation

Помимо этого, манипуляции со строками можно использовать и для других целей:

* Это расширение полезно при увеличении и уменьшении числа.
* Перестановку символов легко адаптировать во время кодирования.
* Изменение регистра, инверсия регистра и заглавные буквы[8].

# 1.3. Описание библиотек, используемых в проекте

В языке python существует множество библиотек модулей, в этой главе я расскажу о тех, которые используются в моем проекте

[**pickle**](https://docs.python.org/3/library/pickle.html#module-pickle)

Модуль [pickle](https://docs.python.org/3/library/pickle.html#module-pickle) реализует бинарные протоколы для сериализации и десериализации структуры объектов Python. «Pickling» — это процесс, при котором иерархия объектов Python преобразуется в поток байтов, а «unpickling» — это обратная операция, при которой поток байтов (из [двоичного файла](https://docs.python.org/3/glossary.html#term-binary-file) или [объекта, подобного байтам](https://docs.python.org/3/glossary.html#term-bytes-like-object) ) преобразуется обратно в иерархию объектов. Pickling (и unpickling) также известны как «serialization», «marshalling» [[ 1 ]](https://docs.python.org/3/library/pickle.html#id7) или «flattening»; однако, чтобы избежать путаницы, здесь используются термины «pickling» и «unpickling»[5].

**PyQt5**

PyQt - это набор инструментов для создания виджетов с графическим интерфейсом. Это интерфейс Python для Qt, одной из самых мощных и популярных кроссплатформенных библиотек с графическим

интерфейсом. PyQt был разработан компанией RiverBank Computing Ltd[6].

**PyQt5.QtWidgets**

**PyQt5.QtWidgets** — это **библиотека на Python, которая содержит большинство виджетов, доступных в PyQt5**. Модуль направлен на создание графических приложений в стиле объектно-ориентированного подхода, где любой компонент интерфейса представлен объектом класса.

**os**

Этот модуль обеспечивает переносимый способ использования функциональности, зависящей от операционной системы. Если вы просто хотите прочитать или записать файл, см. [open()](https://docs.python.org/3/library/functions.html#open), если вы хотите манипулировать путями, см. [os.path](https://docs.python.org/3/library/os.path.html#module-os.path) модуль, а если вы хотите прочитать все строки во всех файлах в командной строке, см. [fileinput](https://docs.python.org/3/library/fileinput.html#module-fileinput) модуль. Для создания временных файлов и каталогов см. [tempfile](https://docs.python.org/3/library/tempfile.html#module-tempfile) модуль, а для высокоуровневой обработки файлов и каталогов см. [shutil](https://docs.python.org/3/library/shutil.html#module-shutil) модуль[7].

**1.4 Программа для разработки графических интерфейсов**

Также для придания более выразительной внешней оболочки приложения, что, несомненно, является важным аспектом разработки, использовалась программа QTdesigner, которая заслуживает особого внимания.

Qt Designer - это мощный и многофункциональный визуальный инструмент, предоставляемый всемирно известным Qt Framework для быстрого, а также эффективного проектирования и построения пользовательских интерфейсов (UI). Этот удивительный инструмент предоставляет интуитивно понятный интерфейс перетаскивания для добавления и настройки различных виджетов, таких как кнопки, метки, текстовые поля и многое другое. В сочетании с PyQt5, который является библиотекой для Python, Qt Designer может значительно упростить и ускорить процесс проектирования пользовательских интерфейсов, а также генерации кода на Python, что в конечном итоге приводит к значительному сокращению времени разработки и усилий, затрачиваемых на этот процесс.

Qt Designer обычно входит в комплект установки Qt framework, что делает его доступным для разработчиков без каких-либо дополнительных затрат. Чтобы успешно использовать Qt Designer с PyQt5, убедитесь, что в вашей системе установлен PyQt5, поскольку это критически важно для корректной работы и взаимодействия между этими инструментами.

# Глава 2.Практическое применение знаний

# 2.1.

Обучение языку программирования Python, безусловно, являлось для меня важным и увлекательным шагом, который я начал три года назад в учебном центре IT-cube. В течение этих лет я прошел несколько этапов обучения и закончил два года учебного процесса – сначала обычный курс, а затем более углубленный, который позволил мне погрузиться в детали и нюансы программирования. О большенстве библиотек, которые широко используются в проекте, я узнал именно там, в увлекательной и познавательной обстановке, что значительно обогатило мой опыт и знания.

Для написания своих программ я изучал перечисленные библиотеки более досконально и сосредоточенно. В результате глубокого анализа и практического опыта я остановился на следующих библиотеках: pickle, PyQt5 и его подмодуль PyQt5.QtWidgets, а также os. Каждая из них сыграла свою уникальную роль в моем процессе разработки.

Библиотеки Pickle и os я использовал для сохранения выбранной даты и её описания в отдельном файле, что, безусловно, является важной частью организации и хранения данных в приложении. С другой стороны, PyQt5 и его подмодуль PyQt5.QtWidgets я использовал для создания интуитивно понятных рабочих кнопок и для работы с progress bar'ом, что делает взаимодействие пользователя с программой более удобным и приятным. Таким образом, я смог реализовать функции, которые значительно улучшили опыт конечных пользователей и повысили эффективность работы программы.

# 2.2.

Для создания простого, но в то же время привлекательного визуального интерфейса я использовал программу QTdesigner. В этом мощном инструменте я смог создать небольшое окно, в котором расположил календарь, таблицу для записи, отсчет в днях и прогресс бар для более наглядного отслеживания прогресса. Это создало удобную и интуитивно понятную среду для пользователя.

После того как дизайн был завершен, я перенес свой проект из QTdesigner в среду разработки PyCharm в виде HTML кода. Этот процесс оказался довольно интересным, и я с интересом наблюдал, как всё это преобразуется. Затем, с помощью различных инструментов, предлагаемых программой, я аккуратно перенес содержимое из HTML кода в полноценный код на Python и начал реализовывать необходимый функционал.

Реализация функционала происходила постепенно и организованно. Сначала я работал над тем, чтобы при выборе даты на календаре автоматически менялось количество отсчета дней до выбранной даты. После этого я сосредоточился на том, чтобы реализовать возможность изменения этого количества в отдельном виджете. И, в последнюю очередь, я занимался созданием прогресс бара, который бы наглядно демонстрировал оставшиеся дни, а также обеспечивал соответствие отсчета правилам русского языка.

Позже, когда основные функции были реализованы, я начал обдумывать, как же лучше сохранить выбранные изменения. У меня был выбор: использовать отдельный файл для хранения данных или же внедрить SQL. SQL (Structured Query Language, или язык структурированных запросов) — это декларативный язык программирования (язык запросов), который во многом становится стандартом для создания, обработки и хранения данных в реляционных базах данных.

В конце концов, мой выбор остановился на методе сохранения данных в отдельном файле. Я пришел к выводу, что для хранения одной выбранной даты и соответствующего описания к ней использование базы данных будет излишним; этот подход показался мне более надежным и удобным, одновременно минимизируя нагрузку на систему.

# ЗАКЛЮЧЕНИЕ

В ходе работы над проектом была достигнута поставленная цель, которая заключалась в создании программы, для более удобного подсчета дней до необходимой даты, и что бы не забыть, какое событие должно произойти. Например, я его используя для отслеживания предзащиты.

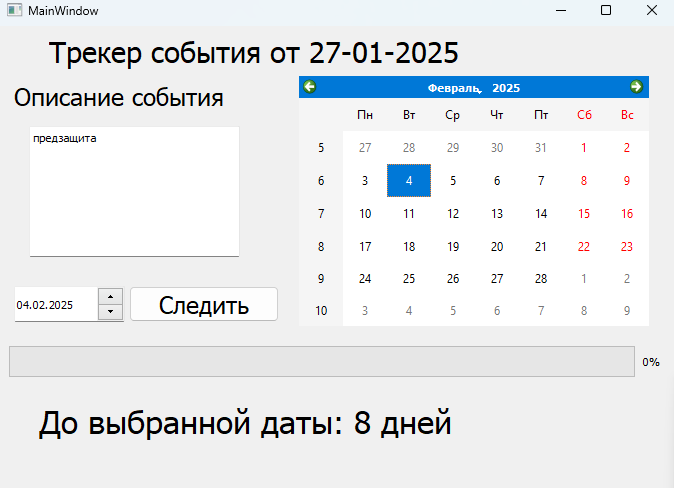


Рис.6. Пример программы

Получивший трекер необходим для людей, которые боятся что-то забыть или упустить. Программа подойдёт всем, ведь каждому из нас так необходимо не забыть что-то важно.

В будущем я бы хотел усовершенствовать программу. Я планирую улучшить визуальную составляющую программы, добавить время и уведомления на рабочий стол.

# СПИСОК ЛИТЕРАТУРЫ