|  |  |
| --- | --- |
| **Изображение выглядит как текст  Автоматически созданное описание** | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | |
| Институт информационных технологий | |
| Кафедра прикладной математики  **КУРСОВАЯ РАБОТА**  по дисциплине  Разработка программных приложений  **Тема курсовой работы**: *Средства для работы с датами*   |  |  | | --- | --- | | Студент группы ИНБО-05-18 | Головко Алексей Николаевич | |  |  | |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) | |  |  | | Руководитель курсовой работы | доцент кафедры ПМ Митина Ольга Алексеевна | |  |  | |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись руководителя) | |  |  | | Работа представлена к защите | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | |  |  | | Допущен к защите | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2020 г. | | |

|  |  |
| --- | --- |
| **Изображение выглядит как текст  Автоматически созданное описание** | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | |
| Институт информационных технологий | |

Кафедра прикладной математики

**ЗАДАНИЕ**

**на выполнение курсовой работы по дисциплине**

Разработка программных приложений

|  |  |
| --- | --- |
| Студент Головко Алексей Николаевич | Группа ИНБО-05-18 |

**Тема работы:** *«Средства для работы с датами»*

**Исходные данные:** среда разработки на языке Python, принципы объектно-ориентированного подхода в проектировании и программировании, стандарты языка Python.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

1. Создание графического дизайна.
2. Реализация и тестирование приложения средствами языка Python.

|  |  |  |  |
| --- | --- | --- | --- |
| **Срок представления к защите курсовой работы:** | | до «\_\_\_» \_\_\_\_\_\_\_\_\_\_2020 г. | |
|  | |  | |
| **Задание на курсовую работу выдал** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | (доцент кафедры ПМ Митина Ольга Алексеевна) |
|  | | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2020 г. | |
| **Задание на курсовую работу получил** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | (Головко А.Н.) |

ОГЛАВЛЕНИЕ

[1 ВВЕДЕНИЕ 4](#_Toc40377025)

[2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc40377026)

[2.1 Модуль Datetime 5](#_Toc40377027)

[2.2 Основы парадигмы структурного программирования 5](#_Toc40377029)

[2.3 Основы парадигмы объектно-ориентированного программирования. 6](#_Toc40377030)

[3 ПРАКТИЧЕСКАЯ ЧАСТЬ 7](#_Toc40377031)

[3.1 Проектное решение 8](#_Toc40377032)

[3.2 Описание структуры курсового проекта для пользователя 9](#_Toc40377034)

[4 Тестирование 10](#_Toc40377035)

[4.1 Пример работы программы 11](#_Toc40377036)

[5 Вывод 12](#_Toc40377038)

[6 СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 13](#_Toc40377039)

[ПРИЛОЖЕНИЕ 14](#_Toc40377040)

1. ВВЕДЕНИЕ

Календарь – это неотъемлемая часть нашей жизни , ведь мы все придерживаемся расписания .Пользуясь календарём , человек , сам того не замечая работает с датами . Мы прибавляем и вычитаем дни , уточняем какой день недели будет в нужную нам дату , считаем разницу между двумя датами.

Данное приложение , выполняет все эти действия . Помимо этого , приложение делает это гораздо быстрее и точнее чем человек .

Целью курсовой работы является создание программного приложения для работы с датами.

Задачи, решаемые в данной курсовой работе:

* анализ научной и методической литературы по проблеме разработки программного приложения;
* разработка и реализация требующихся структур в приложении;
* использование знаний по технологии структурного программирования с использованием современных средств PyCharm;
* обучение качественному оформлению документации.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Одним из плюсов языка Python является огромное количество открытых для любого пользователя библиотек . Этим может похвастаться не каждый язык программирования . Доступность делает Python популярным среди новичков , так как легко можно найти метод , помогающий реализовать поставленные перед программистом задачи

Данная работа – не исключение . В языке программирования Python хорошо реализована работа с датой и временем .

* 1. Модуль datetime

Модуль datetime содержит три основных класса

* Date
* Time
* Datetime

В своей работе я использовал только класс Date. При помощи методов которые содержит этот класс мне удалось обработать вводимую пользователем дату и выполнить все необходимые операции с датой .

* 1. Структурное программирование

Методология разработки программного обеспечения, в основе которой лежит представление программы в виде иерархической структуры блоков. Предложена в 1970-х годах Э. Дейкстрой и др.

В соответствии с данной методологией любая программа строится без использования оператора goto из трёх базовых управляющих структур: последовательность, ветвление, цикл; кроме того, используются подпрограммы.

При этом разработка программы ведётся пошагово, методом «сверху вниз» .Методология структурного программирования появилась как следствие возрастания сложности решаемых на компьютерах задач, и соответственно, усложнения программного обеспечения.

В 1970-е годы объёмы и сложность программ достигли такого уровня, что традиционная (неструктурированная) разработка программ перестала удовлетворять потребностям практики.

Программы становились слишком сложными, чтобы их можно было нормально сопровождать. Поэтому потребовалась систематизация процесса разработки и структуры программ .Методологияструктурной разработки программного обеспечения была признана «самой сильной формализацией 70-х годов*».*

* 1. Основы объектно–ориентированного программирования

Методология программирования основанная на представлении программы в виде совокупности объектов каждый из которых является экземпляром определённого класса , а классы образуют иерархию наследования

В языке Python ООП играет ключевую роль. Даже программируя в рамках структурной парадигмы, вы все равно пользуетесь объектами и классами, пусть даже встроенными в язык, а не созданными лично вами.

Основными понятиями, используемыми в ООП, являются класс, объект, наследование, инкапсуляция и полиморфизм. В языке Python класс равносилен понятию тип данных.

1. ПРАКТИЧЕСКАЯ ЧАСТЬ

Следующим этапом при разработке программного приложения является постановка задачи:

* разработать и реализовать программное приложение, предназначенное для работы с датами
* разработать программное приложение с графическим интерфейсом, демонстрирующее применение структур языка Python в данном проекте.

В следующем разделе будет рассмотрено проектное решение курсовой работы.

* 1. Проектное решение

Для составления проектного решения необходимо выполнить объектную декомпозицию задачи.

Таблица 1

|  |  |
| --- | --- |
| Идентификатор объекта класса | Семантика |
| ourdate | Функция выводит дату во всех нужных нам форматах , а также номер дня недели и день недели |
| Plus1 | Функция которая увеличивает дату на один день |
| plusn | Функция в которой определен размер третьего окна |
| Min1 | Функция , которая уменьшает дату на 1 день |
| minn | Функция для вычисления разницы между датой и числом n |
| delta | Функция для рассчета количество дней между двумя датами |

Таким образом, в таблице представлена вся необходимая информация о создании программного приложения для работы с датами .

* 1. Описание структуры курсового проекта для пользователя

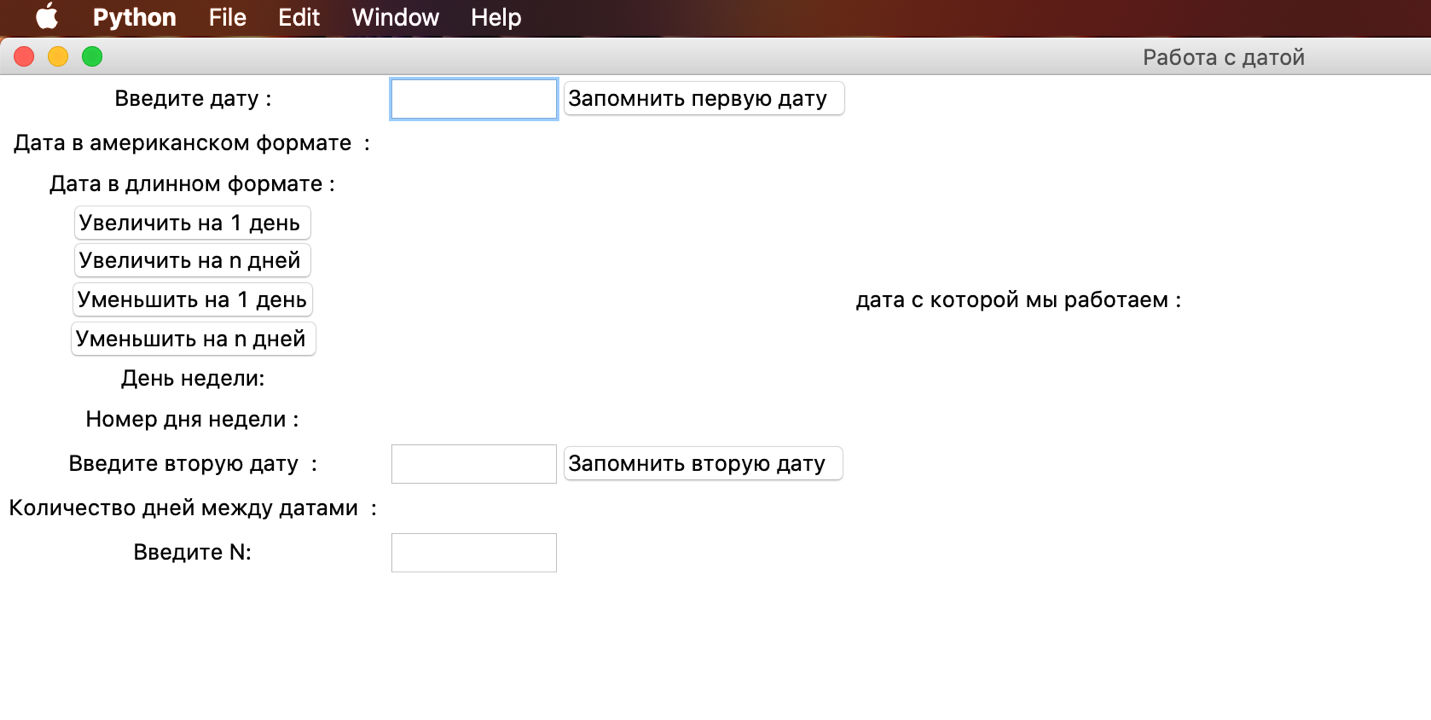
Разработанный проект позволяет производить различные манипуляции с датой . **

Рисунок 1 Окно приложения

Форма, разработанного в ходе курсового проекта программного приложения, позволяет пользователю производить различные манипуляции с датой . В данном блоке:

Поля ввода данных организованы следующим образом:

* Первая дата ;
* Вторая дата;
* Число N ;

Функционал кнопок:

* Выводит день недели и дату в нужных нам форматах (запомнить первую дату);
* Вычисляет разницу между первой и второй датой (запомнить вторую дату );
* Уменьшает дату на 1 день (увеличить на 1 день).
* Увеличивает дату на n дней (увеличить на n дней)
* Уменьшает дату на один день (уменьшить на 1 день)
* Уменьшает дату на n дней (уменьшить на n дней )

В данном разделе представлены основные задачи и цели курсового проекта, а также рассмотрена декомпозиция программного кода и описаны методы классов, которые необходимы для создания программного приложения и дальнейшего его тестирования.

1. Тестирование

В ходе разработки ПО были проведены различные тесты при которых находились и исправлялись некоторые ошибки приложения.

Тестирования производились во время всей разработки приложения, различными пользователями на операционной системе Windows 10. Тестовые запуски показали, что приложение:

• Работоспособно.  
• Понятно и удобно , благодаря простому интерфейсу программы.

• Функционально. Приложение выполнено полностью в соответствии с ТЗ

* 1. Пример работы программы

Пользователь ввел дату , на рисунке ниже видно , что

* Приложение вывело дату в американском формате
* Приложение вывело дату в длинном формате
* После ввода второй даты , приложение посчитало количество дней между датами
* Приложение указало день недели и номер дня недели (1- понедельник , 7 – воскресенье )

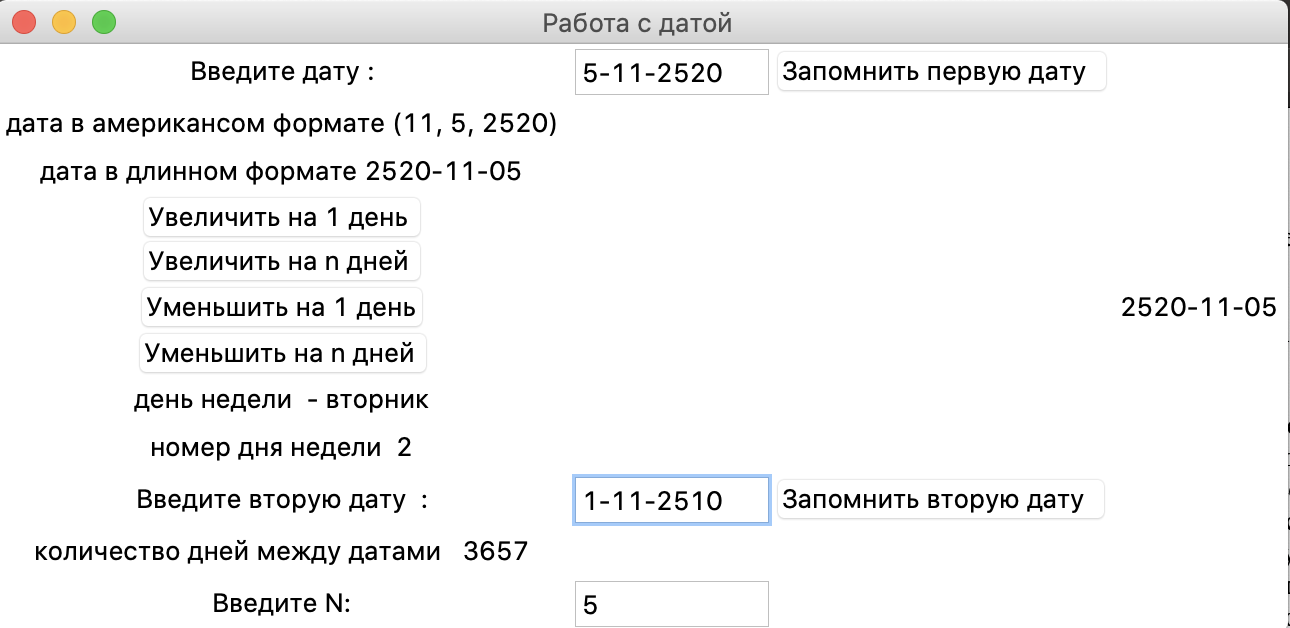


Рис2. Пример работы приложения

1. Вывод

По итогам разработки приложения были выполнены следующие задачи

• Была выбрана программная среда

• Была спроектирована логика и интерфейс приложения.

• Для приложения был разработан интуитивно понятный интерфейс, не вызывающий трудностей в работе.

• Все функции приложения, указанные в техническом задании, были реализованы, и работают без ошибок.

• Были произведены тесты, для оценки стабильности и работоспособности приложения. • Поставленная цель курсовой работы была выполнена.

1. СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ
2. Лутц Марк. Программирование на Python. Издательство «Символ-Плюс», Москва, 2011. – 288с.
3. Официальный сайт Python https://www.python.org.
4. ЕСПД. ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

ПРИЛОЖЕНИЕ

import datetime  
from tkinter import \*  
import re  
def ourdate():  
 ourdate = (inputdate\_text.get())  
 year, month, day = map(int, ourdate.split('-'))  
 date1 = datetime.date(day, month, year)  
 output\_label.configure(text = date1)  
 date3 = month, year, day  
 date2 = "дата в американсом формате {}".format(date3)  
 americanformat\_label.configure(text = date2)  
 date4 = "дата в длинном формате {}".format(date1)  
 longformat\_label.configure(text = date4)  
 k = date1.isoweekday()  
 dayNumber = "номер дня недели {}".format(k)  
 daynumber\_label.configure(text = dayNumber)  
 if k == 1:  
 day\_label.configure(text = " день недели - понедельник")  
 elif k == 2:  
 day\_label.configure(text = "день недели - вторник")  
 elif k == 3:  
 day\_label.configure(text="день недели - среда")  
 elif k == 4:  
 day\_label.configure(text="день недели - четверг")  
 elif k == 5:  
 day\_label.configure(text="день недели - пятница")  
 elif k == 6:  
 day\_label.configure(text="день недели - суббота")  
 elif k == 7:  
 day\_label.configure(text="день недели - воскрвенье")  
def plus1():  
 ourdate = (inputdate\_text.get())  
 year, month, day = map(int, ourdate.split('-'))  
 date1 = datetime.date(day , month, year + 1)  
 output\_label.configure(text = date1)  
def plusn():  
 n = (inputN\_text.get())  
 ourdate = (inputdate\_text.get())  
 year, month, day = map(int, ourdate.split('-'))  
 date1 = datetime.date(day , month, year + int(n))  
 output\_label.configure(text = date1)  
def min1():  
 ourdate = (inputdate\_text.get())  
 year, month, day = map(int, ourdate.split('-'))  
 date1 = datetime.date(day , month, year - 1)  
 output\_label.configure(text = date1)  
def minn():  
 n = (inputN\_text.get())  
 ourdate = (inputdate\_text.get())  
 year, month, day = map(int, ourdate.split('-'))  
 date1 = datetime.date(day, month, year - int(n))  
 output\_label.configure(text=date1)  
def delta():  
 first = (inputdate\_text.get())  
 year, month, day = map(int, first.split('-'))  
 date1 = datetime.date(day, month, year )  
 second = (inputdate1\_text.get())  
 year, month, day = map(int, second.split('-'))  
 date2 = datetime.date(day, month, year)  
 delta = date1 - date2  
 k = delta.days  
 Numbers\_label  
 D = "количество дней между датами {}".format(k)  
 Numbers\_label.configure(text = D)  
  
window = Tk()  
window.title("Работа с датой ")  
inputdate\_label = Label(window , text = "Введите дату :")  
inputdate\_label.grid(column = 0 , row = 0 )  
inputdate\_text = Entry(window , width = 10 )  
inputdate\_text.grid(column = 1 , row = 0 )  
inputdate1\_label = Label(window , text = "Введите вторую дату :")  
inputdate1\_label.grid(column = 0 , row = 9 )  
inputdate1\_text = Entry(window , width = 10 )  
inputdate1\_text.grid(column = 1 , row = 9 )  
americanformat\_label = Label(window , text = "Дата в американском формате :")  
americanformat\_label.grid(column = 0 , row = 1 )  
longformat\_label = Label(window , text = "Дата в длинном формате :")  
longformat\_label.grid(column = 0 , row = 2 )  
output\_label = Label(window , text = "дата с которой мы работаем :")  
output\_label.grid(column = 3 , row = 5)  
btn\_input1 = Button(window, text="Запомнить первую дату " , command = ourdate)  
btn\_input1.grid(column=2, row=0 )  
btn\_input2 = Button(window, text="Запомнить вторую дату " , command = delta)  
btn\_input2.grid(column=2, row=9)  
btn1 = Button(window, text="Увеличить на 1 день " , command = plus1)  
btn1.grid(column=0, row=3)  
btn2 = Button(window, text="Увеличить на n дней " , command = plusn)  
btn2.grid(column=0, row=4)  
btn3 = Button(window, text="Уменьшить на 1 день" , command = min1)  
btn3.grid(column=0, row=5)  
btn4 = Button(window, text="Уменьшить на n дней " , command = minn)  
btn4.grid(column=0, row=6)  
day\_label = Label(window , text = "День недели:")  
day\_label.grid(column = 0 , row = 7)  
daynumber\_label = Label(window , text = "Номер дня недели :")  
daynumber\_label.grid(column = 0, row = 8)  
Numbers\_label = Label(window , text = "Количество дней между датами :")  
Numbers\_label.grid(column = 0, row = 10)  
N\_label = Label(window , text = "Введите N:")  
N\_label.grid(column = 0, row = 11)  
inputN\_text = Entry(window , width = 10 )  
inputN\_text.grid(column = 1 , row = 11 )  
window.mainloop()