**МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Северо-Кавказский федеральный университет» Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №1**

**по дисциплине «Проектирование человеко-машинного интерфейса»**

|  |
| --- |
| Выполнил студент группы ПИЖ-б-о-20-1 |
| Ваньянц И.М. « » 20 г. |
| Подпись студента |
| Работа защищена « » \_20 г. |
| Проверил Мельников С.В.  (подпись) |

Ставрополь 2022

**ХОД РАБОТЫ**

Задание 1.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ВВЕДЕНИЕ

1.Наименование: конвертер температуры.

2.Область применения: облегчение интеллектуальной работы при расчёте физических формул.

ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

1.Документ: лабораторная работа.

2.Организация, утвердившая этот документ, и дата его утверждения: СКФУ, 13.09.2022

3.Наименование темы разработки: Разработка программы для преобразования температур.

НАЗНАЧЕНИЕ РАЗРАБОТКИ

1.Назначение программы: преобразование температур из Цельсия в Фаренгейт и обратно, из Цельсия в Кельвин и обратно, из Фаренгейта в Кельвин и обратно.

ТРЕБОВАНИЯ К ПРОГРАММЕ

1.Требования к функциональным характеристикам: Программа должна позволять осуществить перевод чисел из одной градусной меры в другую.

Исходные данные: целые числа.

Входные и выходные данные: простые и вещественные числа.

2.Требования к надежности: при выполнении операции конвертации не должно возникать ошибок.

3.Условия эксплуатации: не указано

4.Требования к составу и параметрам технических средств: IBM-совместимый персональный компьютер (ПЭВМ), включающий в себя:

1.Процессор Pentium-1000 с тактовой частотой, ГГц – 10, не менее;

2.Оперативную память объемом, Гб – 2, не менее.

5.Требования к информационной и программной совместимости: Программа должна работать автономно под управлением ОС MS Windows версии не ниже 7. Базовый язык программирования – Python.

6.Требования к маркировке и упаковке: не указано

7.Требования к транспортированию и хранению: не указано

8.Специальные требования: Добавьте также возможность использовать температуру в Кельвинах с возможностью преобразования между любыми из трех единиц измерения.

ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

1.Руководство системного программиста: программа состоит из 6 функций, выполняющих операции конвертации, также присутствует 7 программных блоков, которые осуществляют построение программного интерфейса.

1.Руководство оператора: для конвертации определённого числа введите его в левое поле, когда операция ввода завершена нажмите кнопку со стрелочкой для конвертации, результат отобразится в правой части отдела.

СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

1. Разработка функций конвертации.
2. Разработка графического интерфейса:
3. Создание окна и определение его масштаба.
4. Поле для конвертации 1. (F -> C)
5. Поле для конвертации 2. (C -> F)
6. Поле для конвертации 3. (K -> C)
7. Поле для конвертации 4. (K -> F)
8. Поле для конвертации 5. (C -> K)
9. Поле для конвертации 6. (F -> K)
10. Тестирование программы, исправление ошибок и дальнейший ее выпуск.

Срок выполнения: 2 недели.

Задание 2.

ПРОГРАММА

Код программы:

import tkinter as tk  
  
  
def fahrenheit\_to\_celsius():  
 fahrenheit = fahr\_ent\_temperature.get()  
 celsius = (5 / 9) \* (float(fahrenheit) - 32)  
 fahr\_lbl\_result["text"] = f"{round(celsius, 2)} \N{DEGREE CELSIUS}"  
  
  
def celsius\_to\_fahrenheit():  
 celsius = cels\_ent\_temperature.get()  
 fahrenheit = 32 + (float(celsius) \* (9 / 5))  
 cels\_lbl\_result["text"] = f"{round(fahrenheit, 2)} \N{DEGREE FAHRENHEIT}"  
  
  
def kelvin\_to\_celsius():  
 kelvin = KelvCels\_ent\_temperature.get()  
 celsius = float(kelvin) - 273.15  
 KelvCels\_lbl\_result["text"] = f"{round(celsius, 2)} \N{DEGREE CELSIUS}"  
  
  
def kelvin\_to\_fahrenheit():  
 kelvin = KelvFahr\_ent\_temperature.get()  
 fahrenheit = (float(kelvin) - 273.15) \* 9 / 5 + 32  
 KelvFahr\_lbl\_result["text"] = f"{round(fahrenheit, 2)} \N{DEGREE FAHRENHEIT}"  
  
  
def celsium\_to\_kelvin():  
 celsius = CelsKelv\_ent\_temperature.get()  
 kelvin = float(celsius) + 273.15  
 CelsKelv\_lbl\_result["text"] = f"{round(kelvin, 2)} \N{KELVIN SIGN}"  
  
  
def fahrenheit\_to\_kelvin():  
 fahrenheit = FahrKelv\_ent\_temperature.get()  
 kelvin = (float(fahrenheit) - 32) \* 5 / 9 + 273.15  
 FahrKelv\_lbl\_result["text"] = f"{round(kelvin, 2)} \N{KELVIN SIGN}"  
  
  
# Создание окна.  
window = tk.Tk()  
window.geometry("250x350")  
window.title("Конвертер температуры")  
window.resizable(width=False, height=False)  
  
# Блок Фаренгейт -> Цельсий /////////////////////////////////////////////////  
fahr\_frm\_entry = tk.Frame(master=window)  
fahr\_ent\_temperature = tk.Entry(master=fahr\_frm\_entry, width=10)  
fahr\_lbl\_temp = tk.Label(master=fahr\_frm\_entry, text="\N{DEGREE FAHRENHEIT}")  
  
  
fahr\_ent\_temperature.grid(row=1, column=0, sticky="e")  
fahr\_lbl\_temp.grid(row=1, column=1, sticky="w")  
  
  
fahr\_btn\_convert = tk.Button(  
 master=window,  
 text="\N{RIGHTWARDS BLACK ARROW}",  
 command=fahrenheit\_to\_celsius  
)  
fahr\_lbl\_result = tk.Label(master=window, text="\N{DEGREE CELSIUS}")  
  
  
fahr\_frm\_entry.grid(row=1, column=0, padx=10)  
fahr\_btn\_convert.grid(row=1, column=1, pady=10)  
fahr\_lbl\_result.grid(row=1, column=2, padx=10)  
# Блок Фаренгейт -> Цельсий закончен /////////////////////////////////////////  
  
# Блок Цельсий -> Фаренгейт /////////////////////////////////////////////////////////  
cels\_frm\_entry = tk.Frame(master=window)  
cels\_ent\_temperature = tk.Entry(master=cels\_frm\_entry, width=10)  
cels\_lbl\_temp = tk.Label(master=cels\_frm\_entry, text="\N{DEGREE CELSIUS}")  
  
cels\_ent\_temperature.grid(row=2, column=0, sticky="e", pady=20)  
cels\_lbl\_temp.grid(row=2, column=1, sticky="w")  
  
cels\_btn\_convert = tk.Button(  
 master=window,  
 text="\N{RIGHTWARDS BLACK ARROW}",  
 command=celsius\_to\_fahrenheit  
)  
cels\_lbl\_result = tk.Label(master=window, text="\N{DEGREE FAHRENHEIT}")  
  
cels\_frm\_entry.grid(row=2, column=0, padx=10,)  
cels\_btn\_convert.grid(row=2, column=1, pady=10)  
cels\_lbl\_result.grid(row=2, column=2, padx=10,)  
# Окончание блока Цельсий -> Фаренгейт /////////////////////////////////////////////  
  
# Блок Кельвин -> Цельсий ///////////////////////////////////////////////////////////  
KelvCels\_frm\_entry = tk.Frame(master=window)  
KelvCels\_ent\_temperature = tk.Entry(master=KelvCels\_frm\_entry, width=10)  
KelvCels\_lbl\_temp = tk.Label(master=KelvCels\_frm\_entry, text="\N{KELVIN SIGN}")  
  
KelvCels\_ent\_temperature.grid(row=3, column=0, sticky="e", pady=20)  
KelvCels\_lbl\_temp.grid(row=3, column=1, sticky="w")  
  
KelvCels\_btn\_convert = tk.Button(  
 master=window,  
 text="\N{RIGHTWARDS BLACK ARROW}",  
 command=kelvin\_to\_celsius  
)  
KelvCels\_lbl\_result = tk.Label(master=window, text="\N{DEGREE CELSIUS}")  
  
KelvCels\_frm\_entry.grid(row=3, column=0, padx=10,)  
KelvCels\_btn\_convert.grid(row=3, column=1, pady=10)  
KelvCels\_lbl\_result.grid(row=3, column=2, padx=10,)  
# Окончание блока Кельвин -> Цельсий ///////////////////////////////////////////////  
  
# # Блок Кельвин -> Фаренгейт //////////////////////////////////////////////////////  
KelvFahr\_frm\_entry = tk.Frame(master=window)  
KelvFahr\_ent\_temperature = tk.Entry(master=KelvFahr\_frm\_entry, width=10)  
KelvFahr\_lbl\_temp = tk.Label(master=KelvFahr\_frm\_entry, text="\N{KELVIN SIGN}")  
  
KelvFahr\_ent\_temperature.grid(row=4, column=0, sticky="e", pady=20)  
KelvFahr\_lbl\_temp.grid(row=4, column=1, sticky="w")  
  
KelvFahr\_btn\_convert = tk.Button(  
 master=window,  
 text="\N{RIGHTWARDS BLACK ARROW}",  
 command=kelvin\_to\_fahrenheit  
)  
KelvFahr\_lbl\_result = tk.Label(master=window, text="\N{DEGREE FAHRENHEIT}")  
  
KelvFahr\_frm\_entry.grid(row=4, column=0, padx=10,)  
KelvFahr\_btn\_convert.grid(row=4, column=1, pady=10)  
KelvFahr\_lbl\_result.grid(row=4, column=2, padx=10,)  
# Окончание блока Кельвин -> Фаренгейт//////////////////////////////////////////  
  
# Блок Цельсий -> Кельвин ///////////////////////////////////////////////////////////  
CelsKelv\_frm\_entry = tk.Frame(master=window)  
CelsKelv\_ent\_temperature = tk.Entry(master=CelsKelv\_frm\_entry, width=10)  
CelsKelv\_lbl\_temp = tk.Label(master=CelsKelv\_frm\_entry, text="\N{DEGREE CELSIUS}")  
  
CelsKelv\_ent\_temperature.grid(row=5, column=0, sticky="e", pady=20)  
CelsKelv\_lbl\_temp.grid(row=5, column=1, sticky="w")  
  
CelsKelv\_btn\_convert = tk.Button(  
 master=window,  
 text="\N{RIGHTWARDS BLACK ARROW}",  
 command=celsium\_to\_kelvin  
)  
CelsKelv\_lbl\_result = tk.Label(master=window, text="\N{KELVIN SIGN}")  
  
CelsKelv\_frm\_entry.grid(row=5, column=0, padx=10,)  
CelsKelv\_btn\_convert.grid(row=5, column=1, pady=10)  
CelsKelv\_lbl\_result.grid(row=5, column=2, padx=10,)  
# Окончание блока Цельсий -> Кельвин ///////////////////////////////////////////////  
  
# # Блок Фаренгейт -> Кельвин //////////////////////////////////////////////////////  
FahrKelv\_frm\_entry = tk.Frame(master=window)  
FahrKelv\_ent\_temperature = tk.Entry(master=FahrKelv\_frm\_entry, width=10)  
FahrKelv\_lbl\_temp = tk.Label(master=FahrKelv\_frm\_entry, text="\N{DEGREE FAHRENHEIT}")  
  
FahrKelv\_ent\_temperature.grid(row=6, column=0, sticky="e", pady=20)  
FahrKelv\_lbl\_temp.grid(row=6, column=1, sticky="w")  
  
FahrKelv\_btn\_convert = tk.Button(  
 master=window,  
 text="\N{RIGHTWARDS BLACK ARROW}",  
 command=fahrenheit\_to\_kelvin  
)  
FahrKelv\_lbl\_result = tk.Label(master=window, text="\N{KELVIN SIGN}")  
  
FahrKelv\_frm\_entry.grid(row=6, column=0, padx=10,)  
FahrKelv\_btn\_convert.grid(row=6, column=1, pady=10)  
FahrKelv\_lbl\_result.grid(row=6, column=2, padx=10,)  
# Окончание блока Фаренгейт -> Кельвин//////////////////////////////////////////  
  
# Запуск приложения.  
window.mainloop()

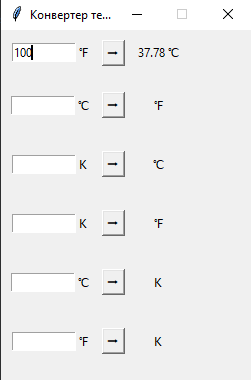


Рисунок 1 – работа первого поля

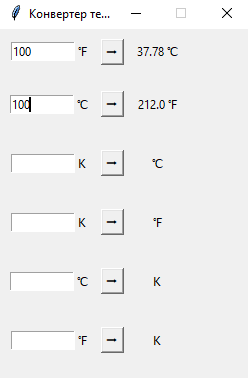


Рисунок 2 – работа второго поля

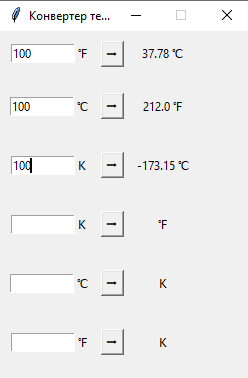


Рисунок 3 – работа третьего поля

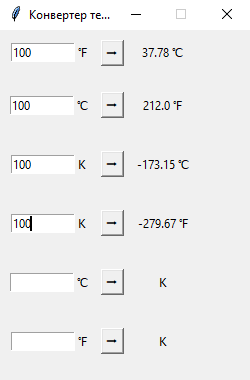


Рисунок 4 – работа четвертого поля

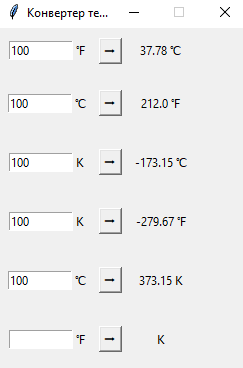


Рисунок 5 – работа пятого поля

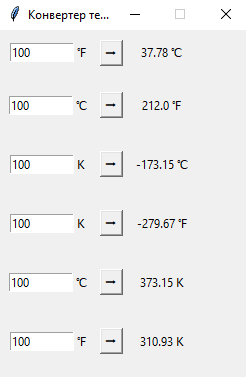


Рисунок 6 – работа шестого поля

**ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Для чего разрабатывается ТЗ - основной документ, содержащий требования заказчика к системе, в соответствии с которыми осуществляется создание и разработка конечного продукта.
2. Какими стандартами регулируется содержимое технического задания? - Техническое задание на программу и программное обеспечение разрабатывается в соответствии с требованиями следующих документов:

– ГОСТ 19.201–78. Единая система программной документации.

Техническое задание. Требования к содержанию и оформлению;

– ГОСТ 34.602–89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.

Основанием для разработки ТЗ чаще всего является договор.

1. Какие существуют стадии и этапы разработки? - Согласно стандарту техническое задание должно содержать следующие разделы:

– введение;

– основания для разработки;

– назначение разработки;

– требования к программе или программному изделию;

– требования к программной документации;

– технико-экономические показатели;

– стадии и этапы разработки;

– порядок контроля и приемки.

1. Раскройте понятие «время восстановления после отказа». - Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.
2. Что должен содержать подраздел «Требования к функциональным характеристикам» раздела «Требования к программе или программному изделию»? - В подразделе «Требования к информационной и программной совместимости» должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой.

При необходимости должна обеспечиваться защита информации и

программ.

1. Обязательно ли присваивать условное обозначение темы разработки? – Нет, не обязательно.