



亞洲大學
ASIA UNIVERSITY

**Midterm Project Report
Advanced Computer Programming**

Finance Tracker with Python

Student Name : Benedictus Keanu Prabowo

Student ID : 112021187

Teacher : DINH-TRUNG VU

2024-04

1) Introduction

3. Github

- 1) **Personal Github Account:** KeanuBenedictus
- 2) **Group Github Account:** FullyMed
- 3) **Group Project Repository:** Amigo
- 4) **List of submitted files:**
 - [Benedictus Keanu 112021187.py](#)

4. Topic

The topic of Web Services for Finance Tracker is chosen because we want to make it easier for users to manage and analyzing their personal finances.

5. Project Overview

The income management system is designed to assist users in managing their income data efficiently. It provides functionalities to add, remove, edit, view income records, calculate statistics, and visualize income distribution. Users can interact with the system through a menu based interface, They Input choices to perform specific actions like adding, removing, editing, viewing income, calculating statistics, plotting income distribution, or exiting the program.

2) Implementation

2. Class

The Class 'Money' serves as a blueprint for creating objects that represent a collection of income data and provide methods for manipulating and analyzing that data. as well as for calculating statistics and visualizing the income distribution.

2.1. Def init

This method is the constructor of the class. It initializes the income attribute as an empty list when a new instance of the Money class is created.

2.2. Def add_income

This method takes an income entry as input and appends it to the 'income' list. It provides feedback to the user by printing "Income Added Successfully".

2.3. Def remove_income

This method removes the income entry at the specified index from the income list. It checks if the index is valid (within the range of the list) before removing the entry. If the index is invalid, it prints "Invalid index".

2.4. Def edit_income

This method updates the income entry at the specified index with a new income value provided by the user. Similar to remove_income, it checks if the index is valid before performing the update.

2.5. Def view_income

This method displays all income entries along with their corresponding indices. It iterates over the 'income' list, printing each entry with its index.

2.6. Def total_income

This method calculates the sum of all the incomes stored in the 'income' list attribute

2.7. Def average_income

This method calculates the average income from the list of incomes stored in 'income'. It sums up all the incomes and then divides the sum by the total number of incomes to compute the average. If there are no incomes stored, it returns 0 to avoid division by zero and to signify that there's no income data available.

2.8. **Def max_income**

This method returns the highest income value stored in the list of incomes. It makes use of Python's built-in 'max()' function to find the maximum value in the list.

2.9. **Def min_income**

this method returns the lowest income value stored in the list of incomes. It uses Python's built-in min() function to find the minimum value in the list.

2.10. **import matplotlib.pyplot as plt**

It imports the pyplot module from the matplotlib library and assigns it the alias plt. This allows you to access functions and classes from the pyplot module using the plt alias, making it shorter and easier to use in your code. In here, matplotlib.pyplot is used to create a histogram of income distribution in the plot_income_distribution method of the 'Money' class. This method plots a histogram of the income data stored in the income list attribute of the 'Money' class.

2.11. **Def plot_income_distribution**

This method generates a histogram plot using Matplotlib to visualize the distribution of income data stored in the 'income' list attribute of the 'Money' class instance

Functionality:

1. It uses Matplotlib's 'plt.hist()' function to create a histogram.
2. The income data stored in the 'self.income' list is used as the input data for the histogram.
3. The 'bins' parameter specifies the number of bins (or intervals) into which the data is divided.
4. The 'color' parameter sets the color of the bars in the histogram.
5. The 'edgecolor' parameter sets the color of the edges of the bars.
6. It adds labels to the x-axis and y-axis using 'plt.xlabel()' and 'plt.ylabel()' respectively.
7. It sets the title of the plot using 'plt.title()'.
8. It enables grid lines on the plot with plt.grid(True).
9. Displays the plot using 'plt.show()'.

3) Results

6. Output

Menu:

1. Add Income
2. Remove Income
3. Edit Income
4. View Income
5. Calculate Statistics
6. Plot Income Distribution
7. Exit

Choose: 1/2/3/4/5/6/7: 1

Enter the income: 5000

Income Added Successfully

Menu:

1. Add Income
2. Remove Income
3. Edit Income
4. View Income
5. Calculate Statistics
6. Plot Income Distribution
7. Exit

Choose: 1/2/3/4/5/6/7: 1

Enter the income: 10000

Income Added Successfully

Menu:

1. Add Income
2. Remove Income
3. Edit Income
4. View Income

5. Calculate Statistics
6. Plot Income Distribution
7. Exit

Choose: 1/2/3/4/5/6/7: 1

Enter the income: 15000

Income Added Successfully

Menu:

1. Add Income
2. Remove Income
3. Edit Income
4. View Income
5. Calculate Statistics
6. Plot Income Distribution
7. Exit

Choose: 1/2/3/4/5/6/7: 4

Income:

0: 5000.0

1: 10000.0

2: 15000.0

Menu:

1. Add Income
2. Remove Income
3. Edit Income
4. View Income
5. Calculate Statistics
6. Plot Income Distribution
7. Exit

Choose: 1/2/3/4/5/6/7: 2

Enter the index of the income to remove: 1

Income Removed Successfully

Menu:

1. Add Income
2. Remove Income
3. Edit Income
4. View Income
5. Calculate Statistics
6. Plot Income Distribution
7. Exit

Choose: 1/2/3/4/5/6/7: 4

Income:

0: 5000.0

1: 15000.0

Menu:

1. Add Income
2. Remove Income
3. Edit Income
4. View Income
5. Calculate Statistics
6. Plot Income Distribution
7. Exit

Choose: 1/2/3/4/5/6/7: 3

Enter the index of the income to edit: 1

Enter the new income: 10000

Income Edited Successfully

Menu:

1. Add Income
2. Remove Income
3. Edit Income
4. View Income
5. Calculate Statistics
6. Plot Income Distribution
7. Exit

Choose: 1/2/3/4/5/6/7: 5

Total Income: 15000.0

Average Income: 7500.0

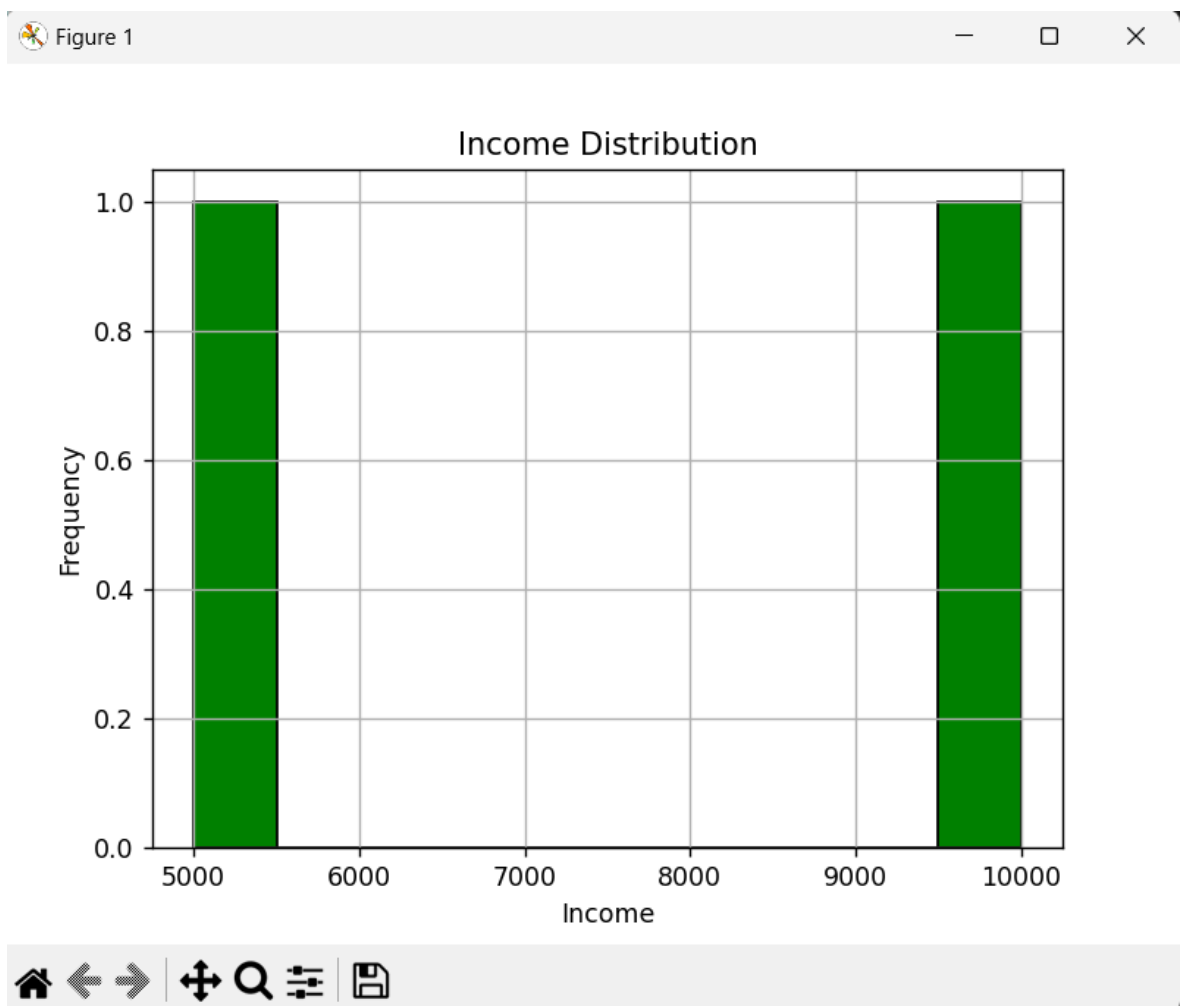
Maximum Income: 10000.0

Minimum Income: 5000.0

Menu:

1. Add Income
2. Remove Income
3. Edit Income
4. View Income
5. Calculate Statistics
6. Plot Income Distribution
7. Exit

Choose: 1/2/3/4/5/6/7: 6



Menu:

1. Add Income
2. Remove Income
3. Edit Income
4. View Income
5. Calculate Statistics
6. Plot Income Distribution
7. Exit

Choose: 1/2/3/4/5/6/7: 7

4) Conclusions

The code defines a `Money` class that facilitates managing income data. It allows users to add, remove, edit, view, and analyze income. Additionally, it offers functionalities to calculate statistics such as total income, average income, maximum income, and minimum income. Moreover, it enables users to visualize the income distribution through a histogram. The code is structured to provide a user-friendly menu-driven interface, allowing users to interact with the income data conveniently.