**КУРСОВИЙ ПРОЕКТ**

з дисципліни **« Програмування »**

на тему**: «Конвертер валют»**

Студента(ки) 3 курсу, групи 31І

ПІБ Дзюби Владислава Ігоровича

Керівник ст. викл. Шаригін Олександр Анатолійович

(вчене звання, науковий ступінь, прізвище та ініціали)

Національна оцінка

Кількість балів: оцінка ECTS

Члени комісії: к. п. н., доцент Габрусєв В.Ю.

ст. викладач Устименко О.Б.

к. т. н., ст. викладач Шаригін О.А.

Студент ПІБ



(підпис)

**Зміст**

[ВСТУП 3](#_Toc156336956)

[Актуальність теми 3](#_Toc156336957)

[Мета і завдання дослідження 3](#_Toc156336958)

[Об’єкт та предмет дослідження 3](#_Toc156336959)

[Методи дослідження 3](#_Toc156336960)

[Завдання дослідження **Помилка! Закладку не визначено.**](#_Toc156336961)

[Структура роботи 4](#_Toc156336962)

[РОЗДІЛ 1: ОГЛЯД ПРОБЛЕМИ ТА ІСНУЮЧІ СПОСОБИ ВИРІШЕННЯ 5](#_Toc156336963)

[1.1 Постановка задачі 5](#_Toc156336964)

[1.2 Огляд проблеми та аналіз публікацій 5](#_Toc156336965)

[1.3 Огляд існуючих рішень 5](#_Toc156336966)

[Висновки до I розділу 5](#_Toc156336967)

[РОЗДІЛ 2: ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ 7](#_Toc156336968)

[2.1 Огляд та обґрунтування технологій для розробки програмного засобу 7](#_Toc156336969)

[2.2 Алгоритм роботи програмного засобу 7](#_Toc156336970)

[2.3 Інструкція використання 11](#_Toc156336971)

[ЗАГАЛЬНІ ВИСНОВКИ 13](#_Toc156336972)

[ВИКОРИСТАНІ ДЖЕРЕЛА 14](#_Toc156336973)

[ДОДАТКИ 15](#_Toc156336974)

# ВСТУП

## Актуальність теми

Сучасна економічна ситуація та необхідність здійснення валютних операцій надає велике значення програмам обміну валют у швидкому доступі. З метою оптимізації таких операцій та надання зручного інструменту для користувачів розробляється програмний засіб обміну валют.

## Мета і завдання дослідження

Мета проєкту – створення програмного засобу для здійснення валютних операцій, з можливістю вибору валют із простим та зрозумілим інтерфейсом.

Завданням дослідження є:

1. Аналіз валютного ринку та визначення переліку валют, з якими буде працювати програма.
2. Аналіз існуючих рішень, які виконують функцію конвертації валют.
3. Розробка програмного засобу для взаємодії з обміном валют.
4. Тестування програми.

## Об’єкт та предмет дослідження

Об'єктом дослідження є процес автоматизованоъ конверсії валют між банками та клієнтами. Предмет дослідження - розробка програмного продукту для здійснення валютного обміну із консольним інтерфейсом.

## Методи дослідження

1. Аналіз існуючих підходів до автоматизованої конверсії валют.

2. Аналіз результатів тестування розробленого застосунку.

**Особистий внесок автора**

Всі основні результати роботи було розроблено особисто автором, а саме: розроблено програму яка надає користувачам простий та зручний консольний інтерфейс для здійснення обміну валют за оптимальним курсом.

## Структура роботи

Робота складається з 19 сторінок, а саме зі вступу, двох розділів та їх висновків, загальних висновків, списку використаних джерел та додатків.

# РОЗДІЛ 1: ОГЛЯД ПРОБЛЕМИ ТА ІСНУЮЧІ СПОСОБИ ВИРІШЕННЯ

## Постановка задачі

Задача полягає у створенні програми для легкого здійснення валютних операцій, за допомогою простої користувацької програми.

Аналіз вже існуючих програмних як онлайн так і офлайн методів для обміну валюти. Знаходження та порівняння курсів валют різних банків; за основу було взято курс валют саме від “monobank’у”. Для демонстрації роботи програми обміну грошових одиниць, взято чотири валюти: Долар (USD), Євро (EUR), Польські злоті (PLN), Турецькі ліри (TRY) та п’ята як основна Гривня (UAH).

## 1.2 Огляд проблеми та аналіз публікацій

Валютний обмін – це актуальна та значуща проблема в сучасному економічному середовищі. Задача полягає в створенні зручного та ефективного інструменту для валютних операцій, який враховує різноманіття потреб користувачів та надає надійність операцій. Аналіз цих джерел допомагає визначити ключові аспекти валютного обміну, розкриває проблеми, що існують у наявних рішеннях та сприяє обґрунтуванню актуальності розробки нового підходу.

## 1.3 Огляд існуючих рішень

На ринку існує кілька способів для проведення валютного обміну, такі як банки, кантори та валютчики. Проведено аналіз присутніх рішень, зокрема:

1. **Банк [**[**5**](#_ВИКОРИСТАНІ_ДЖЕРЕЛА_1)**]:**
   * *Переваги:* Безпека, гарантована державою; надійність.
   * *Недоліки:* Невигідний офіційний курс валют; обмеження по лімітах; лімітований час роботи банків.
2. **Кантора / Обмінний пункт [**[**6**](#_ВИКОРИСТАНІ_ДЖЕРЕЛА_1)**]:**
   * *Переваги:* Безпека і надійність; наявність валюти.
   * *Недоліки:* Не приймають старі або пошкоджені купюри; лімітований час роботи.
3. **Валютчики:**
   * *Переваги:* Постійна наявність іноземної валюти; можливість обміну у різний час.
   * *Недоліки:* Можливе шахрайство, фальшиві купюри, обрахунок.

## Висновки до I розділу

У першому розділі курсової роботи було висвітлено актуальність обраної теми та проведено аналіз проблеми, а також розглянуто існуючі підходи до її вирішення. Мета та завдання дослідження були чітко сформульовані, визначаючи напрямки подальших досліджень. Об'єкт і предмет дослідження були чітко визначені, встановлюючи межі дослідження.

У контексті методів дослідження вказано, які підходи та методи були використані для збору та аналізу даних, а також формування висновків під час виконання роботи. Завдання дослідження були конкретизовані, визначаючи, що саме планується вирішити за допомогою даного дослідження.

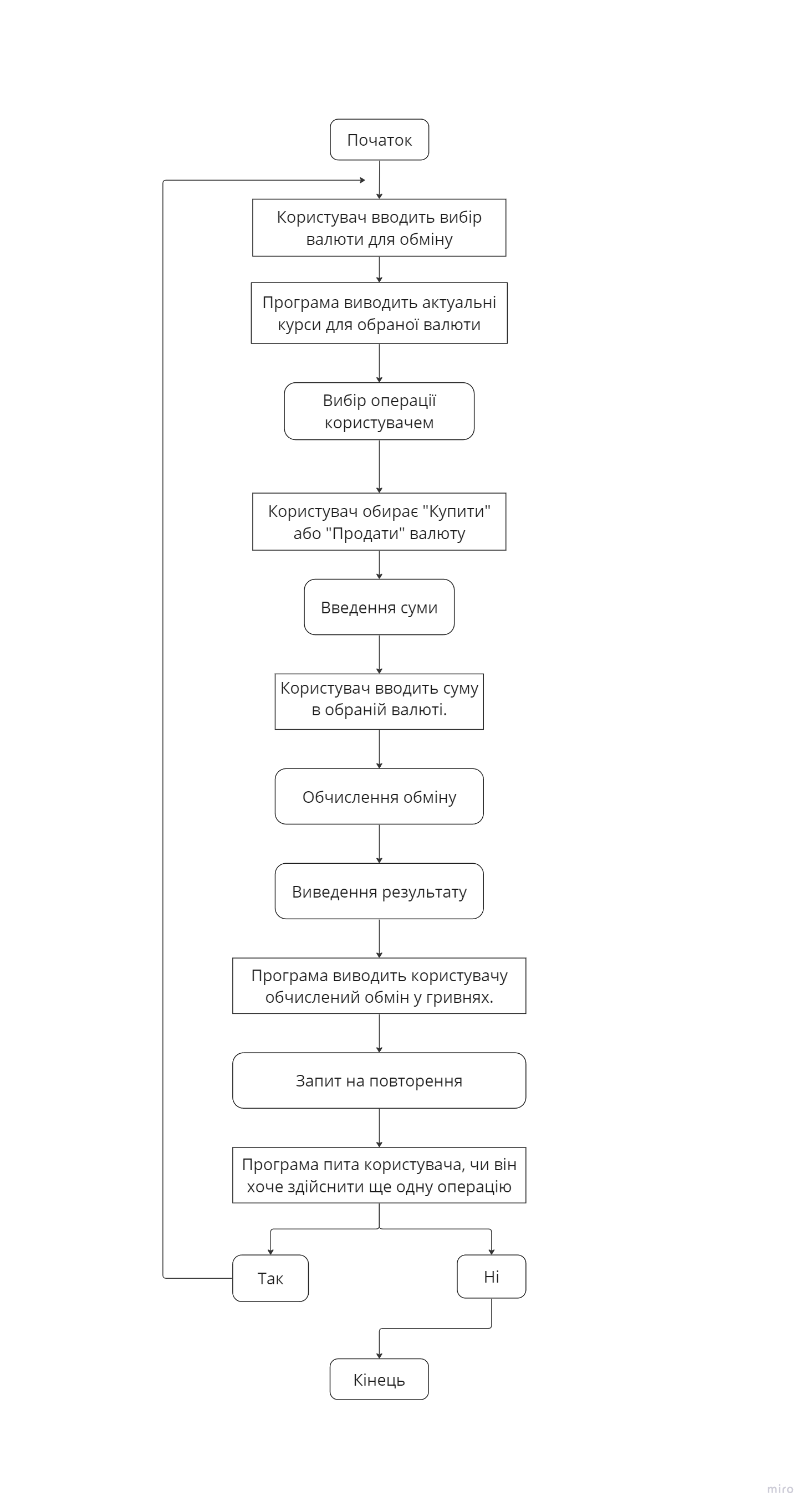
# РОЗДІЛ 2: ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

## 2.1 Огляд та обґрунтування технологій для розробки програмного засобу

1. Операційна система на якій було розроблено застосунок – Windows 11;
2. База даних – не використовується. У програмі записані константні значення курсу купівлі та продажу до кожної валюти (USD, EUR, PLN, TRY) згідно даним від monobank;
3. Мова програмування – Python;
4. Програмний інтерфейс – консоль; подалі можна вдосконалити програму додавши візуальний інтерфейс.

## 2.2 Алгоритм роботи програмного засобу

1. **Алгоритм роботи ПЗ:**

****

1. **Опис використаних стандартних функцій:**

* **print()**: Виведення на екран інформації про курси купівлі та продажу валюти.
* **float()**: Зчитування введеного користувачем значення, перетворення його в тип float.
* **input():** Введення користувачем значень при купівлі чи продажу валюти.
* **round()**: Застосування вбудованої функції для округлення до двох знаків після коми при розрахунку суми для купівлі валюти.

1. **Опис використаних функцій сторонніх бібліотек:**У даному коді не використовуються сторонні бібліотеки. Всі функції базуються на стандартних засобах мови програмування.
2. **Опис глобальних констант та змінних:**

Глобальні константи:

* **usd\_buy, usd\_sell, eur\_buy, eur\_sell, pln\_buy, pln\_sell, try\_buy, try\_sell**: Курси купівлі та продажу для різних валют.

Глобальні змінні:

* **currency\_choice, currency\_choice\_name, currency\_buy\_rate, currency\_sell\_rate**: Змінні, які використовуються для вибору валюти та збереження пов'язаних з нею параметрів.

1. **Опис власних функцій, об’єктів, методів та їх призначення:**

* **display\_rates(buy\_rate, sell\_rate, currency\_name)**: Виводить на екран курси купівлі та продажу для конкретної валюти.
* **get\_float\_input(prompt)**: Забезпечує користувача можливістю вводу числового значення, перевіряє правильність введення.
* **perform\_exchange(buy\_rate, sell\_rate, currency\_name)**: Реалізує процес обміну валют, дозволяючи користувачеві вибирати опції купівлі та продажу.
* **exit\_program()**: Завершує роботу програми.

1. **Навести основні (ключові) фрагменти коду:**

Фрагмент коду програми де показані всі основні функції:

# Основний цикл програми

while True:

# Вибір валюти користувачем

while True:

print("\n1. USD \n2. EUR \n3. PLN \n4. TRY \nSelect currency you want to Buy or Sell: ")

currency\_choice = input()

# Перевірка введення

if currency\_choice.lower() == "end":

exit\_program() # Виходить з програми, якщо користувач вводить "end"

if currency\_choice.isdigit() and 1 <= int(currency\_choice) <= 4:

break

else:

print(f"\nInvalid choice! Choose a number from 1 to 4.")

currency\_choice = int(currency\_choice)

# Визначення обраної валюти та виведення її курсів

if currency\_choice == 1:

display\_rates(usd\_buy, usd\_sell, "US Dollar")

currency\_choice\_name = "USD"

currency\_buy\_rate = usd\_buy

currency\_sell\_rate = usd\_sell

elif currency\_choice == 2:

display\_rates(eur\_buy, eur\_sell, "Euro")

currency\_choice\_name = "EUR"

currency\_buy\_rate = eur\_buy

currency\_sell\_rate = eur\_sell

elif currency\_choice == 3:

display\_rates(pln\_buy, pln\_sell, "Polish zloty")

currency\_choice\_name = "PLN"

currency\_buy\_rate = pln\_buy

currency\_sell\_rate = pln\_sell

elif currency\_choice == 4:

display\_rates(try\_buy, try\_sell, "Turkish lira")

currency\_choice\_name = "TRY"

currency\_buy\_rate = try\_buy

currency\_sell\_rate = try\_sell

# Виклик функції для обміну валют

perform\_exchange(currency\_buy\_rate, currency\_sell\_rate, currency\_choice\_name)

# Питання користувачеві, чи бажає він виконати ще одну операцію обміну

another\_exchange = input("Do you want to perform another currency exchange? (yes/no): ")

if another\_exchange.lower() != 'yes':

exit\_program() # Виходить з програми, якщо користувач не бажає виконати ще одну операцію обміну

В Додатку 1 даний код викладено в повному обсязі, посилання на репозиторій GitHub [[4](#_ВИКОРИСТАНІ_ДЖЕРЕЛА_1)].

## 2.3 Інструкція використання

1. Вибір валюти користувачем, шляхом введення в консолі відповідної цифри, де 1 – USD (долар), 2 – EUR (євро), 3 – PLN (польська злота), 4 – TRY (турецька ліра).  
   Якщо користувач вводить цифру яка не дорівнює числам від одного до чотирьох, то програма виводить повідомлення про те, що треба ввести число від одиниці до четвірки.
2. Програмою виводиться повідомлення про актуальний курс грошової одиниці на купівлю та продаж.
3. Користувачу пропонується вибір, чи він хоче купити чи продати валюту. На що, користувачу треба написати слово “Buy” для купівлі або слово “Sell” для продажу валюти.  
   У випадку якщо юзер неправильно вводить одне зі слів, програма повідомляє користувачу, що треба конкретно ввести одне зі слів “Buy” або “Sell”; і немає значення як вона його впише, чи всі букви великі чи маленькі, програма переводить всі введені значення у малий регістр.
4. Після обраної операції (купівлі чи продажу), користувачу задається питання, яку кількість обраної валюти (крок 1) він хоче купити чи продати, в залежності від його відповіді на попередньому етапі.  
   Якщо ж вводиться від’ємне число або символи програма просить записати додатне число без символів.
5. В залежності від вибору клієнта, у консолі друкується сума яку треба заплатити користувачу в гривнях (UAH) якщо той обрав купівлю (крок 3) або сума яку він отримає у випадку обраного варіанту – продаж (крок 3).
6. Перед особою виникає питання, чи бажає він провести ще один обмін валюти. Якщо користувач вводить “Yes”, то програма повертається на перший крок з можливістю обрати одну з запропонованих грошових одиниць. У випадку якщо користувач ввів “No” або будь що, що не є варіантом “Yes”, то програма припиняє роботу.
7. На протязі всіх етапів програми, окрім четвертого, виконується функція яка працює як “хрестик” закриття програм. Тобто, якщо користувач вводить слово “end” на етапах з першого по третій, то програма закривається. Це було зроблено більше для зручності автора програми, для швидкого тестування програми.

**Висновки до II розділу**

У другому розділі було розглянуто саме процес розробки, основні елементи коду та інструкції, що до роботи з програмою яка здійснює валютний обмін. Логіка роботи програми включає в себе введення та обробку даних, вивід результатів, а також перевірки на коректність введених даних. Програма передбачає можливість завершення роботи за бажанням користувача та надає зручний інтерфейс для повторних операцій.

# ЗАГАЛЬНІ ВИСНОВКИ

Під час розробки був створений програмний засіб для валютного обміну із консольним інтерфейсом. Незважаючи на те, що програма ще не досягла ідеальності, вона є доброю основою для прототипу, до якого можна легко додати графічний інтерфейс для поліпшення зручності використання користувачем. Крім того, у майбутньому можливо вдосконалити програмний код для автоматичного отримання актуальних значень валютних курсів з офіційних банківських API. Також можна розглянути можливість зберігання останніх значень курсів і оновлення їх раз на годину, щоб користувач мав доступ до актуальної інформації навіть при відсутності Інтернету. Крім того, є можливість імпорту програми на різні платформи та використання її на онлайн-ресурсах. Усі ці можливості сприяють розширенню та поліпшенню функціоналу програми, роблячи її більш гнучкою та корисною для користувачів у валютних операціях.

# ВИКОРИСТАНІ ДЖЕРЕЛА

1. Курс: Програмування, 3 СОІ та 2 ІСТ (22-23 н.р.) | УДУ.ФМІФ. УДУ.ФМІФ. URL: <https://moodle.fmif.udu.edu.ua/course/view.php?id=939>
2. Курс валют Monobank. I.UA. URL: <https://finance.i.ua/bank/115/>
3. Python Tutorial. W3Schools. URL: <https://www.w3schools.com/python/>
4. GitHub - vladkus24/Course-Work. URL: <https://github.com/vladkus24/Course-Work>
5. Офіційний курс гривні щодо іноземних валют. Національний банк України. URL: <https://bank.gov.ua/ua/markets/exchangerates/?utm_source=site&utm_medium=ukrainian_city&utm_campaign=article>
6. Мінфін - все про фінанси: новини, курси валют, банки. Мінфін. URL: <https://minfin.com.ua/?utm_source=site&utm_medium=ukrainian_city&utm_campaign=article>

# ДОДАТКИ

**Додаток 1: Код програмного засобу.**

usd\_buy = 38.0000

usd\_sell = 38.4497

eur\_buy = 41.5000

eur\_sell = 42.1496

pln\_buy = 9.7409

pln\_sell = 9.9545

try\_buy = 1.2994

try\_sell = 1.3550

def display\_rates(buy\_rate, sell\_rate, currency\_name):

    print(f"\n{currency\_name} buying rate is {buy\_rate} and selling rate is {sell\_rate}")

def get\_float\_input(prompt):

    while True:

        try:

            user\_input = float(input(prompt))

            if user\_input >= 0:

                return user\_input

            else:

                print("Please enter a non-negative number.")

        except ValueError:

            print("Please enter a valid number.")

def perform\_exchange(buy\_rate, sell\_rate, currency\_name):

    while True:

        print(f"\nDo you want to Buy or Sell {currency\_name}? (Type 'Buy' or 'Sell')")

        buy\_sell\_choice = input()

        if buy\_sell\_choice.lower() == "end":

            exit\_program()

        if buy\_sell\_choice.lower() == "buy":

            bank\_budget = get\_float\_input(f"\nHow much {currency\_name} do you want to buy? ")

            exchange\_buy = round(sell\_rate \* bank\_budget, 2)

            print(f"\nSo, you need to pay {exchange\_buy} UAH.")

            break

        elif buy\_sell\_choice.lower() == "sell":

            client\_budget = get\_float\_input(f"\nHow much {currency\_name} do you want to sell? ")

            exchange\_sell = round(buy\_rate \* client\_budget, 2)

            print(f"\nSo, you will get: {exchange\_sell} UAH.")

            break

        else:

            print("Choose one of the options: Buy or Sell.")

def exit\_program():

    print("\nExiting the program.")

    exit()

while True:

    while True:

        print("\n1. USD \n2. EUR \n3. PLN \n4. TRY \nSelect currency you want to Buy or Sell: ")

        currency\_choice = input()

        if currency\_choice.lower() == "end":

            exit\_program()

        if currency\_choice.isdigit() and 1 <= int(currency\_choice) <= 4:

            break

        else:

            print(f"\nInvalid choice! Choose a number from 1 to 4.")

    currency\_choice = int(currency\_choice)

    if currency\_choice == 1:

        display\_rates(usd\_buy, usd\_sell, "US Dollar")

        currency\_choice\_name = "USD"

        currency\_buy\_rate = usd\_buy

        currency\_sell\_rate = usd\_sell

    elif currency\_choice == 2:

        display\_rates(eur\_buy, eur\_sell, "Euro")

        currency\_choice\_name = "EUR"

        currency\_buy\_rate = eur\_buy

        currency\_sell\_rate = eur\_sell

    elif currency\_choice == 3:

        display\_rates(pln\_buy, pln\_sell, "Polish zloty")

        currency\_choice\_name = "PLN"

        currency\_buy\_rate = pln\_buy

        currency\_sell\_rate = pln\_sell

    elif currency\_choice == 4:

        display\_rates(try\_buy, try\_sell, "Turkish lira")

        currency\_choice\_name = "TRY"

        currency\_buy\_rate = try\_buy

        currency\_sell\_rate = try\_sell

    perform\_exchange(currency\_buy\_rate, currency\_sell\_rate, currency\_choice\_name)

    another\_exchange = input("Do you want to perform another currency exchange? (yes/no): ")

    if another\_exchange.lower() != 'yes':

        exit\_program()