Звіт з практичної роботи No7

Тема роботи: Моделювання циклічних процесів

Виконав: Кравченко Сергій, ТЦР-12

Дата: 02.06.2025

Мета роботи: Опанувати практичні навички роботи з циклами для моделювання

повторюваних процесів у Python. Закріпити вміння використовувати цикли

для вирішення типових практичних задач, розвинути навички

алгоритмічного мислення.

Хід роботи:

Постановка завдання:

Завдання 1. «Банківський депозит»

Код програми: def bank\_deposit():

initial\_amount = float(input("Введіть початкову суму вкладу: "))

interest\_rate = float(input("Введіть процентну ставку (річних): "))

target\_amount = float(input("Введіть бажану кінцеву суму: "))

years = 0

current\_amount = initial\_amount

while current\_amount < target\_amount:

years += 1

current\_amount \*= (1 + interest\_rate / 100)

print(f"Рік {years}: {current\_amount:.2f}")

print(f"Потрібно років: {years}")

bank\_deposit()

Приклад роботи програми:

Введіть початкову суму вкладу: 1000

Введіть процентну ставку (річних): 5

Введіть бажану кінцеву суму: 1500

Рік 1: 1050.00

Рік 2: 1102.50

Рік 3: 1157.63

Рік 4: 1215.51

Рік 5: 1276.28

Рік 6: 1340.10

Рік 7: 1407.10

Рік 8: 1477.46

Рік 9: 1551.33

Потрібно років: 9

Завдання 2. «Гра вгадай число з обмеженням спроб»

Код програми:

import random

def guess\_number():

secret\_number = random.randint(1, 100)

print("Я загадав число від 1 до 100. У тебе 7 спроб!")

for attempt in range(1, 8):

guess = int(input(f"Спроба {attempt}. Твоя відповідь: "))

if guess < secret\_number:

print("Загадане число більше!")

elif guess > secret\_number:

print("Загадане число менше!")

else:

print(f"Вітаю! Ти вгадав за {attempt} спроб!")

return

print(f"На жаль, ти не вгадав. Загадане число було {secret\_number}.")

guess\_number()

Приклад роботи програми:

Я загадав число від 1 до 100. У тебе 7 спроб!

Спроба 1. Твоя відповідь: 50

Загадане число менше!

Спроба 2. Твоя відповідь: 30

Загадане число більше!

Спроба 3. Твоя відповідь: 45

Загадане число менше!

Спроба 4. Твоя відповідь: 40

Загадане число більше!

Спроба 5. Твоя відповідь: 42

Вітаю! Ти вгадав за 5 спроб!

Завдання 3. «Виведення простих чисел у заданому діапазоні»

Код програми:

def is\_prime(n):

if n < 2:

return False

for i in range(2, int(n\*\*0.5) + 1):

if n % i == 0:

return False

return True

def prime\_numbers():

lower = int(input("Введіть нижню межу діапазону: "))

upper = int(input("Введіть верхню межу діапазону: "))

primes = []

for num in range(lower, upper + 1):

if is\_prime(num):

primes.append(num)

if primes:

print("Прості числа у заданому діапазоні:", primes)

else:

print("У заданому діапазоні немає простих чисел.")

prime\_numbers()

Приклад роботи програми:

Введіть нижню межу діапазону: 10

Введіть верхню межу діапазону: 30

Прості числа у заданому діапазоні: [11, 13, 17, 19, 23, 29]

Завдання 4. «Обчислення факторіала числа з перевіркою вводу»

Код програми:

def factorial():

while True:

try:

n = int(input("Введіть ціле невід'ємне число: "))

if n < 0:

print("Число має бути невід'ємним!")

continue

break

except ValueError:

print("Будь ласка, введіть ціле число!")

result = 1

process = []

if n == 0:

print("0! = 1")

return

for i in range(1, n + 1):

result \*= i

process.append(str(i))

print(f"{n}! = {'\*'.join(process)} = {result}")

factorial()

Приклад роботи програми:

Введіть ціле невід'ємне число: 5

5! = 1\*2\*3\*4\*5 = 120

Введіть ціле невід'ємне число: -2

Число має бути невід'ємним!

Введіть ціле невід'ємне число: abc

Будь ласка, введіть ціле число!

Введіть ціле невід'ємне число: 4

4! = 1\*2\*3\*4 = 24

Завдання 5. «Моделювання росту популяції бактерій»

Код програми:

def bacteria\_growth():

initial\_bacteria = 10

growth\_rate = 20 # 20%

max\_population = int(input("Введіть максимальну кількість бактерій: "))

hours = 0

current\_population = initial\_bacteria

while current\_population < max\_population:

hours += 1

current\_population \*= (1 + growth\_rate / 100)

print(f"Година {hours}: {int(current\_population)} бактерій")

print(f"Досягнуто максимальної популяції за {hours} годин.")

bacteria\_growth()

Приклад роботи програми:

Введіть максимальну кількість бактерій: 1000

Година 1: 12 бактерій

Година 2: 14 бактерій

Година 3: 17 бактерій

Година 4: 20 бактерій

Година 5: 24 бактерій

Година 6: 28 бактерій

Година 7: 33 бактерій

...

Година 25: 985 бактерій

Година 26: 1182 бактерій

Досягнуто максимальної популяції за 26 годин.

Висновки

Робота демонструє, що цикли є потужним інструментом для автоматизації повторюваних процесів у програмуванні. Опанування їх використання дозволяє ефективно розв’язувати задачі з реального життя, від фінансів до наукових досліджень.