Завдання до розрахункової роботи № 1.
Консольна програма для розв'язування прикладної задачі.

Завдання 1. Розробити консольну програму для розв'язання прикладної обчислювальної задачі згідно варіанта. Реалізувати розв'язування прямої фізичної задачі (обчислення за формулою) та обернених задач (знаходження одного з параметрів формули за заданим результатом і значеннями інших параметрів) у вигляді окремих функцій. В основному коді реалізувати введення потрібних даних і вивід розв'язків кожної із запрограмованих задач.

Завдання 2. Використовуючи функцію для розв'язання задачі з завдання 1 (прямої задачі) та функціонал модулів ***matplotlib*** та ***numpy*** створити програму для побудови графіка залежності результату обчислень від зміни одного з параметрів. Відрізок та крок зміни параметра а також фіксовані значення інших параметрів вводяться користувачем з клавіатури. Реалізувати побудову графіка у вигляді окремих функцій для кожного з параметрів, які містить формула.

Завдання 3. Доповнити програму функціями для побудови графіків для обернених задач. Усі створені в ході виконання завдань функції розмістити в окремому модулі. В основній програмі реалізувати можливість вибору користувачем розв'язуваної задачі у режимі діалогу. Для реалізації вибору використати механізм у вигляді масиву покажчиків на функції. Для отримання рішення користувача вивести на консоль відповідне повідомлення з переліком пунктів для вибору та зчитати вибраний користувачем пункт.

Звіт про виконання розрахункової роботи повинен містити *титульний аркуш*, розв'язок кожної задачі, що включає *умову задачі*, *код програми* та *результати її виконання* для деякого набору вхідних даних.

Варіанти завдань

1. Знайти максимальну висота польоту тіла випущеного з початковою швидкістю v_0 під кутом $0^0 < \alpha < 90^0$ до горизонту: $h_{\max} = \dfrac{(v_0 \sin\alpha)^2}{2g}$.
2. Визначити тиск стовпа висотою h рідини, що має густину ρ: $p = \rho g h$.
3. Знайти потенціальну енергію V м^3 води, піднятих греблею на висоту h м.: $W_n = \rho V g h$. Густину води вважати рівною 1000 кг/ м^3.
4. За законом Архімеда розрахувати умову плавання тіла масою m та об'ємом V, зануреного у рідину з густиною ρ.
5. Визначити число ядер радіоактивної речовини, що розпалися за період часу t, якщо відомо сталу радіоактивного розпаду λ та початкове число ядер N_0.

6. Обчислити величину роботи з переміщення тіла на відстань S, виконаною силою F, прикладеною у напрямку переміщення.
7. Встановити кількість теплоти потрібну для нагрівання m кг заданої речовини на t градусів.
8. Встановити величину сили F, яку потрібно прикласти до пружини заданої жорсткості, щоб стиснути її на вказану величину.
9. Знайти нормальне прискорення тіла, що рухається по колу радіуса R зі сталою лінійною швидкістю v.
10. Визначити швидкість поширення звуку в рідині з модулем усестороннього тиску K та густиною ρ.
11. Обчислити ємність плоского конденсатора з площею пластин S та відстанню d між ними.
12. Визначити кількість теплоти, що утворюється від згоряння m кг заданої речовини з питомою теплотою згоряння q Дж/кг.
13. Обчислити потужність механізму, що може виконати роботу A за час t.
14. Знайти кінетичну енергію тіла масою m, що рухається зі швидкістю v: $W_k = \frac{mv^2}{2}$.
15. Визначити відстань, яку пролетить тіло випущене з початковою швидкістю v_0 під кутом $0^0 < \alpha < 90^0$ до горизонту: $h_{\max} = \frac{v_0{}^2 \sin(2\alpha)}{g}$.
16. Визначити силу струму в провіднику за законом Ома: $I = \frac{U}{R}$.

Приклад виконання роботи

Варіант 16. Визначити силу струму в провіднику за законом Ома: $I = \frac{U}{R}$.

Завдання 1:

1. Реалізовуємо функцію для розв'зування прямої задачі. Задача, очевидно полягає у знаходженні сили струму за заданими значеннями напруги та опору. Усі величини можуть бути представлені дійсними числами, тому для вводу даних використовуємо функцію ***float***. Орієнтовний код функції та програми:

```
# функція для обчислення сили струму за заданими
# значеннями напруги та опору (пряма задача)
def amperage(u, r):
    i = u / r
    return i
# основний код програми
print('Обчислення сили струму за законом Ома.')
U = float(input('введіть величину напруги (Вольт)'))
R = float(input('введіть величину опору (Ом)'))
```

```
I = amperage(U, R);
print('Сила струму: %10.3f'%I)
```

2. До поставленої задачі можна сформулювати дві обернені задачі:

- за заданою силою струму та напругою визначити опір: $R = \frac{U}{I}$.

- за заданою силою струму та опором кола визначити напругу: $U = R \cdot I$.

Реалізовуємо ще дві функції для їх розв'язання, та перевіряємо їх роботу в основному коді. Код до завдання 1 може бути таким:

```
# функція для обчислення сили струму за заданими
# значеннями напруги та опору (пряма задача)
def amperage(u, r):
    i = u / r
    return i
# функція для обчислення опору за заданими
# значеннями сили струму та напруги (обернена задача 1)
def resistance(u, i):
    r = u / i
    return r
# функція для обчислення напруги за заданими
# значеннями сили струму та опору (обернена задача 2)
def voltage(r, i):
    u = r * i
    return u
# основний код програми
print('Обчислення сили струму за законом Ома.')
U = float(input('введіть величину напруги (В): '))
R = float(input('введіть величину опору (Ом): '))
I = amperage(U, R)
print('Сила струму: %10.3f Ампер'%I)
# Обернена задача 1
print('Обчислення електричного опору за законом Ома.')
U = float(input('введіть величину напруги (В): '))
I = float(input('введіть величину струму (А): '))
R = resistance(U, I)
print('Опір: %10.3f Ом'%R)
# Обернена задача 2
print('Обчислення напруги за законом Ома.')
I = float(input('введіть величину струму (А): '))
R = float(input('введіть величину опору (Ом): '))
U = voltage(R, I)
print('Напруга: %10.3f Вольт'%U)
```

Завдання 2:

1. Використовуємо написану в попередньому завданні функцію для знаходження сили струму:

```
def amperage(u, r):
    i = u / r
    return i
```

2. Для першого графіка візьмемо в якості змінного параметра напругу U та побудуємо графік залежності сили струму від напруги при сталому значенні опору R кола. На другому - побудуємо графік залежності сили струму від опору при сталому значенні напруги U. Спочатку реалізуємо побудову графіка в основному коді. Орієнтовний код програми:

```
import math
import numpy
import matplotlib.pyplot as plot
# функція для обчислення сили струму за заданими
# значеннями напруги та опору (пряма задача)
def amperage(u, r):
    i = u / r
    return i
# графік залежності сили струму від напруги
# деяке фіксоване значення опору
R = 20
# діапазон для зміни параметра напруги
Ua = 5
Ub =10
U = numpy.linspace(Ua,Ub,100)
y = []
for x in U:
    y.append(amperage(x,R))
I = numpy.array(y)

plot.plot(U, I)
plot.show()

# графік залежності сили струму від напруги
# деяке фіксоване значення опору
U = 20
# діапазон для зміни параметра напруги
Ra = 5
Rb =10
R = numpy.linspace(Ra,Rb,100)
y = []
```

```
for x in R:
    y.append(amperage(U,x))
I = numpy.array(y)

plot.plot(R, I)
plot.show()
```

3. Винесемо код побудови графіків в окремі функції. Побудову графіків в основному коді залишимо з тими ж параметрами. Орієнтовний код програми:

```
import numpy
import matplotlib.pyplot as plot
# функція для обчислення сили струму за заданими
# значеннями напруги та опору (пряма задача)
def amperage(u, r):
    i = u / r
    return i
# функція для побудови графіка залежності
# сили струму від напруги при
# заданому значенні опору
def plotAmperageByVoltage(R,Ua,Ub):
    U = numpy.linspace(Ua,Ub,100)
    y = []
    for x in U:
        y.append(amperage(x,R))
    I = numpy.array(y)
    plot.plot(U, I)
    plot.show()

# функція для побудови графіка залежності
# сили струму від напруги
# деяке фіксоване значення опору
def plotAmperageByResistance(U,Ra,Rb):
    R = numpy.linspace(Ra,Rb,100)
    y = []
    for x in R:
        y.append(amperage(U,x))
    I = numpy.array(y)
    plot.plot(R, I)
    plot.show()

# Основний код
R0 = 20
# діапазон для зміни параметра напрунги
U1 = 5
U2 =10
```

```
# вивід графіка залежності сили струму від напруги
plotAmperageByVoltage(R0,U1,U2)
# графік 2
U0 = 20
# діапазон для зміни параметра напрунги
R1 = 5
R2 =10
# вивід графіка залежності сили струму від опору
plotAmperageByResistance(U0,R1,R2)
```

Завдання 3:

1. Скористаємося кодом створеним у **завданні 2** з функціями для побудови двох графіків. Для повного виконання умови завдання слід додати в модуль ще дві пари функцій: для побудови графіків залежності напруги від опору та сили струму і опору від напруги та сили струму, а також потрібні для цього функції з завдання 1. Тобто, модуль, який в прикладі нижче містить 3 функції, насправді повинен містити 9, так само і меню в основному коді, – не 2 пункти, а 6.

2. Збережемо функції з попереднього завдання в окремому Python-файлі з довільною назвою (в нашому прикладі ***plotmodule***). Назва створеного файлу в основному коді буде використовуватися як назва модуля у імпорт-виразі, він повинен міститися в тому ж пакеті (папці), що і файл з основним кодом. Орієнтовний код модуля:

```
import numpy
import matplotlib.pyplot as plot
# функція для обчислення сили струму за заданими
# значеннями напруги та опору (пряма задача)
def amperage(u, r):
    i = u / r
    return i
# функція для побудови графіка залежності
# сили струму від напруги при
# заданому значенні опору
def plotAmperageByVoltage(R,Ua,Ub):
    U = numpy.linspace(Ua,Ub,100)
    y = []
    for x in U:
        y.append(amperage(x,R))
    I = numpy.array(y)
    plot.plot(U, I)
    plot.show()
# функція для побудови графіка залежності
# сили струму від напруги
```

```
# деяке фіксоване значення опору
def plotAmperageByResistance(U,Ra,Rb):
    R = numpy.linspace(Ra,Rb,100)
    y = []
    for x in R:
        y.append(amperage(U,x))
    I = numpy.array(y)
    plot.plot(R, I)
    plot.show()
```

3. В основному коді організовуємо вивід пунктів меню, вибір їх користувачем шляхом вводу відповідної назви, ввід користувачем потрібних параметрів та побудову графіка:

```
# підключення модуля
import plotmodule
# код основної програми
selected = input('Оберіть тип графіка для '
                 'побудови:\n IR - залежність сили '
                 'струму від опору\n IU - залежність'
                 'сили струму від напруги \n>>>>')
# в готовій програмі меню міститиме 6 пунктів
if selected == 'IU':
    R = float(input('Задайте значення'
                    ' опору R:\n'))
    U1 = float(input('Задайте початкове'
                     ' значення зміни напруги U1:\n'))
    U2 = float(input('Задайте кінцеве'
                     ' значення зміни напруги U2:\n'))
    plotmodule.plotAmperageByVoltage(R, U1, U2)
if selected == 'IR':
    U = float(input('Задайте значення'
                    ' напруги U:\n'))
    R1 = float(input('Задайте початкове'
                     ' значення зміни опору R1:\n'))
    R2 = float(input('Задайте кінцеве'
                     ' значення зміни опору R2:\n'))
    plotmodule.plotAmperageByResistance(U, R1, R2)
# тут слід додати код виклику решти
# функцій з відповідними умовами
```

Завдання до розрахункової роботи № 2.

Розробка візуального інтерфейсу користувача до програм для розв'язування прикладних задач.

Завдання. Використовуючи засоби бібліотеки Tkinter розробити віконний візуальний інтерфейс користувача до власного варіанту **лабораторної роботи № 14** (в якості альтернативи можна скористатися проєктом з *лабораторної роботи № 10*, де замість бази даних MySQL використовується зберігання даних у файлі формату csv).

- На початку роботи програма повинна відображати вікно зі списком усіх даних, що містяться в базі.
- Вікно має містити елементи керування (кнопки з текстовими полями, випадні списки і т. п.) для пошуку даних відповідно до варіанту завдання (див. л.р. № 14 (10)).
- Результати пошуку відображати в цьому ж вікні на місці загального списку.
- Додати на вікно кнопку для повернення списку до початкового стану (виводу усіх даних).
- Додати кнопку для запису нових даних в базу. По натисканні цієї кнопки має з'являтися нове вікно в якому вводяться дані та підтверджується їх ввід. Після вводу даних програма повинна повертатися до основного вікна, оновлюючи список даних.

Рекомендована література:

1. Програмування числових методів мовою Python: підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий; за ред. А. В. Анісімова. – К.: Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с.
2. Збірник задач з дисципліни "Інформатика і програмування" / Вакал Є. С., Личман В. В., Обвінцев О. В., Бублик В. В., Довгий Б. П., Попов В. В. -2-ге видання, виправлене та доповнене –К.: ВПЦ "Київський університет", 2006.– 94 с.
3. Караванова Т.П. Основи алгоритмізації та програмування: 777 задач з рекомендаціями та прикладами: Навч. посіб. Доп. та випр. – К.: Генеза, 2006. – 288 с.
4. Копей В. Б. Мова програмування Python для інженерів і науковців: навчальний посібник / В. Б. Копей – Івано-Франківськ : ІФНТУНГ, 2019. – 272 с.
5. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
6. Креневич А.П. Python у прикладах і задачах. Частина 1. Структурне програмування Навчальний посібник із дисципліни "Інформатика та програмування" – К.: ВПЦ "Київський Університет", 2017. – 206 с.
7. Кукурудз С.Ф.Процюк В.Р., Ваврик Т.О. Збірник задач з програмування. Навчальний посібник. – Івано-Франківськ: Факел, 2005. – 247с.
8. Основи науки про дані [Електронний ресурс]: навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення» / М. А. Новотарський; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 5,5 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 294 с.
9. Збірник задач з основ алгоритмізації та програмування. Навчальний посібник з курсів «Обчислювальна техніка та програмування», «Інформатика», «Основи інформатики і програмування», для студентів технічних спеціальностей для першого (бакалаврського) рівня освіти / С. М. Ментинський, Я. М. Пелех. – Львів: Видавництво ТзОВ «Колір ПРО», 2023. – 320 с.
10. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник для студ. спеціальності 122 "Комп’ютерні науки", спеціалізації "Інформаційні технології в біології та медицині" / А. В. Яковенко ; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 195 с.

Навчальне видання

Когут Ігор Васильович

Ментинський Сергій Мирославович

ЛАБОРАТОРНИЙ ПРАКТИКУМ З ПРОГРАМУВАННЯ НА PYTHON.

Лабораторний практикум з курсу
«Основи інформатики і програмування, частина 2»
спеціальності 105 – “Прикладна фізика та наноматеріали”
першого (бакалаврського) рівня освіти

Комп’ютерне верстання: – *С. Ментинський*

Обкладинка розроблена з використанням
ресурсів Freepik.com

Підписано до друку 1.12.2023 р.
Формат 60*84/16. Гарнітура Times New Roman.
Папір офсетний. Друк офсетний. Ум. друк. арк. 9,3.
Наклад 100 прим. Зам. № 2964

ТзОВ «Компанія “Манускрипт”»
вул. Руська, 16/3, м. Львів, 79008
тел./факс: (032) 235-60-00.
Свідоцтво про внесення суб’єкта видавничої справи до державного
реєстру видавців, виготівників і розповсюджувачів видавничої продукції
серія ДК № 3628 від 19. 11. 2009 р.