МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

***Кафедра комп’ютерної інженерії та інформаційних технологій***

***Відділення інженерії програмного забезпечення***

**Звіт з теми № 3**

з курсу «Тренінг-курс з основ програмування»

Малишевої Валерії Олексіївни

студента групи **2П-23**

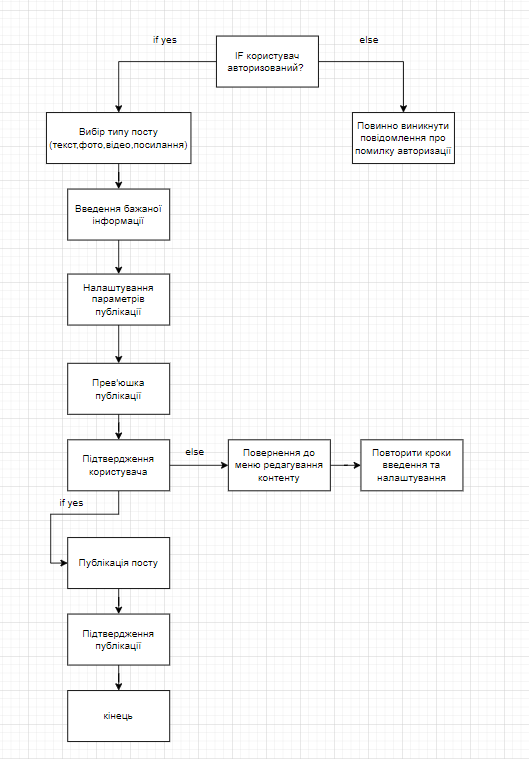
номер у списку підгрупи - 13

Викладач Марченко С. В.

Черкаси-2025

**1.Аналіз постановки задачі**

Визначте та запишіть передумови, постумови та кроки реалізації для визначеного відповідно до Вашого варіанту завдання. Забезпечте відповідність Вашого опису характеристикам алгоритма. Описуйте кроки в форматі постумови = назва\_дії(передумови) з описом внутрішньої логіки за допомогою блоксхем або псевдокодів.

****

**2. Задачі з прийняттям рішень(1, 5, 14, 20):**

1.Припустимо, розумна програма може не вмикати будильник, щоб людина виспалась на вихідних чи у відпустці. Розробіть додаток, який буде приймати на вхід від користувача позначки про вихідний та відпустку, а виводитиме рішення, чи вмикати будильник. Вмикати його не слід, якщо день вихідний або людина у відпустці.

5.Тривалість місяців коливається від 28 до 31 дня.Напишіть програму, яка зчитуватиме назву місяця(рядок). Потім додаток має вивести кількість днів у введеному місяці. Для лютого слід вивести «28 або 29 днів». У випадку некоректного вводу програма має продемонструвати повідомлення щодо цього.

14.Поширеною думкою є те, що 1 рік собачого життя еквівалентний 7 рокам життя людини. Проте це просте правило перестає виконуватись, коли собака стає дорослою (приблизно в 2 роки). У результаті часто вважають, що краще брати перші 2 роки життя собаки еквівалентними 21 рокам життя людини, а подальші роки – чотирьом людським рокам. Напишіть програму, яка реалізує перетворення років життя собаки в роки життя людини за даними правилами. Ваш додаток має виводити попередження, коли вводиться від’ємне число.

20.Мобільний оператор пропонує три інтернет-тарифи для населення:

| Тариф 1000 | 1000 Мб на місяць за 20 грн, позатарифно:  1Мб = 0.05 грн. |
| --- | --- |
| Тариф 2000 | 2000 Мб на місяць за 35 грн, позатарифно:  1Мб = 0.04 грн. |
| Тариф 5000 | 5000 Мб на місяць за 85 грн., позатарифно:  1 Мб = 0.02 грн. |

Напишіть програму, яка повинна отримувати цифру-тариф та кількість витрачених мегабайтів, а виводити підсумковий рахунок за місяць. Якщо використовується тариф 1000 або 2000, виведіть додатково інформацію про те, скільки б платили користувачі, якби перейшли на більш дорогий тариф.

**Лістинг коду до задачі №1(Завдання 2)**

**Лістинг 1.1(Функція для визначення,чи вихідний,чи відпустка)**

*def need\_set\_bydilbnuk(is\_vuhidnuy, is\_vidpystka):*

*if is\_vuhidnuy or is\_vidpystka:*

*return False*

*return True*

**Лістинг 1.2(Функція для зчитування даних введених користувачем)**

*def vvod\_dannih(promt):*

*while True:*

*user\_input = input(promt).strip().lower()*

*if user\_input in ['так', 'ні']:*

*return user\_input == 'так'*

*print("Введіть 'так' або 'ні'")*

**Лістинг 1.3(Функція для прийняття рішення чи вмикати будильник чи ні на основі введених користувачем даних та запуск програми)**

*def main():*

*print("Чи треба вмикати будильник:")*

*is\_vuhidnuy = vvod\_dannih("У вас сьогодні вихідний?")*

*is\_vidpystka = vvod\_dannih("У вас сьогодні відпустка?")*

*set\_bydilbnuk = need\_set\_bydilbnuk(is\_vuhidnuy, is\_vidpystka)*

*if set\_bydilbnuk:*

*print("Будильник встановлено")*

*else:*

*print("Будильник вимкнено")*

*if \_\_name\_\_ == "\_\_main\_\_":*

*main()*

**Лістинг коду до задачі №5(Завдання 2)**

**Лістинг коду 2.1(Функція що приймає аргумент ‘***misyacb*’ (назву місяця що введе користувач)**)**

*def get\_dni\_misyacya(misyacb):*

*dni\_misyacya = { #словник з ключами - місяцями*

*"січень": 31,*

*"лютий": "28 або 29 днів",*

*"березень": 31,*

*"квітень": 30,*

*"травень": 31,*

*"червень": 30,*

*"липень": 31,*

*"серпень": 31,*

*"вересень": 30,*

*"жовтень": 31,*

*"листопад": 30,*

*"грудень": 31*

*}*

*month = misyacb.lower()#на всякий випадок перевід до нижнього реєстру*

*if month in dni\_misyacya: #перевірка чи введений місяць є у словнику*

*return f"У місяці {month} {dni\_misyacya[month]} днів."*

*else:*

*return "Введіть коректну назву."*

**Лістинг коду 2.2(Функція що виконує основну логіку програми)***def main():*

*print("Програма для визначення кількості днів у місяці.")*

*misyacb\_input = input("Введіть назву місяця: ").strip()*

*result = get\_dni\_misyacya(misyacb\_input)*

*print(result)*

*if \_\_name\_\_ == "\_\_main\_\_":*

*main()*

**Лістинг коду до задачі №14(Завдання 2)**

**Лістинг коду 3.1(Функція з одним аргументом ‘***sobaka\_years***’ що приймає кількість років собаки)**

*def sobaka\_to\_lyduna\_years(sobaka\_years):*

*if sobaka\_years < 0:# перевірка чи введене число від’ємне*

*return "Помилка: Введено від'ємне число. Будь ласка, введіть додатне число."*

*elif sobaka\_years <= 2:#Перевірка чи введене число <= 2*

*return sobaka\_years \* 10.5 # Перші 2 роки собаки = 21 рік людини*

*else:*

*return 21 + (sobaka\_years - 2) \* 4 # кожні +2 роки собаки = 4 роки людини*

**Лістинг коду 3.2(Функція що виконує основну логіку)**

*def main():*

*try:#обробка всіх можливих помилок*

*pes\_years = float(input("Введіть кількість років життя собаки: "))#зчитування введеного числа*

*lyduna\_years = sobaka\_to\_lyduna\_years(pes\_years)#Виклик функції sobaka\_to\_lyduna\_years з передачею введеного значення pes\_years. Результат зберігається у змінній lyduna\_years.*

*if isinstance(lyduna\_years, str):#Перевірка чи результат є рядком*

*print(lyduna\_years)*

*else:#якщо результат не є рядком то виконується :*

*print(f"{pes\_years} років життя собаки ≈ {lyduna\_years} років життя людини.")*

*except ValueError:#помилка,якщо було введено не число*

*print("Помилка: Будь ласка, введіть числове значення.")*

*if \_\_name\_\_ == "\_\_main\_\_":#Запуск коду*

*main()*

**Лістинг коду до задачі №20(Завдання 2)**

**Завдання було написано завдяки гпт(він надав кроки вирішення та стак бібліотек для використання(щоб написати код на рівні мідла),а код написаний мною)**

**Лістинг коду 4.1(Імпорт бібліотек)**

*from dataclasses import dataclass #Імпорт dataclass, який дозволяє створювати класи з автоматичною реалізацією методів \_\_init\_\_, \_\_repr\_\_ та інших.*

*from typing import Optional, Dict #Імпорт типів Optional (для вказівки, що значення може бути None) та роботи зі словниками.*

**Лістинг коду 4.2(Клас Tarif)**

*@dataclass #Декоратор, автоматично генерує методи для класу*

*class Tarif:*

*stala\_cina: float #Атрибут, зберігає фіксовану вартість тарифу*

*mb\_scho\_vhodyatb: int #зберігає кількість мегабайтів, які входять у тариф*

*doplata\_za\_mb: float #зберігає вартість одного мегабайта поза тарифом*

*nastypnuy\_tarif: Optional[int] = None #зберігає наступний (більш дорогий) тариф*

**Лістинг коду 4.3(Словник TARIFU)**

*TARIFU: Dict[int, Tarif] = {#Оголошення словника, де ключі — це назви тарифів (1000, 2000, 5000), а значення — об'єкти класу Tarif*

*1000: Tarif(stala\_cina=20, mb\_scho\_vhodyatb=1000, doplata\_za\_mb=0.05, nastypnuy\_tarif=2000),*

*2000: Tarif(stala\_cina=35, mb\_scho\_vhodyatb=2000, doplata\_za\_mb=0.04, nastypnuy\_tarif=5000),*

*5000: Tarif(stala\_cina=85, mb\_scho\_vhodyatb=5000, doplata\_za\_mb=0.02)*

*}*

**Лістинг коду 4.4()**

*def pidrahynok\_cheky(tarif: int, used\_mb: int) -> float:#приймає тариф та кількість витрачених мегабайтів,а повертає загальну вартість*

*if used\_mb < 0:#Перевірка від’ємності витрачених мегабайтів*

*raise ValueError("Кількість витрачених мегабайтів не може бути від'ємною.")#Вивід помилки якщо умова виконалась*

*if tarif not in TARIFU:#Перевірка чи є введений тариф у словнику*

*raise ValueError("Невірний тариф. Будь ласка, оберіть 1000, 2000 або 5000.")*

*tarif\_data = TARIFU[tarif]#Отримання даних про тариф зі словника*

*extra\_mb = max(used\_mb - tarif\_data.mb\_scho\_vhodyatb, 0) # Позатарифні МБ (не менше 0)*

*totalbnuy\_rahynok = tarif\_data.stala\_cina + extra\_mb \* tarif\_data.doplata\_za\_mb*

*#Обчислення загальної вартості: фіксована вартість + вартість позатарифних мегабайтів*

*return totalbnuy\_rahynok#Повернення загальної вартості*

**Лістинг коду 4.5(kraschuy\_rahynok)**

*def kraschuy\_rahynok(teperishniy\_tarif: int, vukorustani\_mb: int) -> Optional[int]:*

*#функція що перевіряє, чи вигідніше перейти на наступний тариф*

*if teperishniy\_tarif not in TARIFU:#Перевірка, чи поточний тариф існує в словнику*

*return None#None,якщо умова виконується*

*teperishniy\_tarif\_data = TARIFU[teperishniy\_tarif]#Отримання даних про поточний тариф*

*nastypnuy\_tarif = teperishniy\_tarif\_data.nastypnuy\_tarif#Отримання наступного тарифу*

*if nastypnuy\_tarif and nastypnuy\_tarif in TARIFU:#Перевірка, чи існує наступний тариф*

*next\_tariff\_data = TARIFU[nastypnuy\_tarif]#Отримання даних про наступний тариф*

*teperishnuy\_rahynok = pidrahynok\_cheky(teperishniy\_tarif, vukorustani\_mb)#Обчислення вартості поточного тарифу*

*maubytnya\_cina = pidrahynok\_cheky(nastypnuy\_tarif, vukorustani\_mb)#Обчислення вартості наступного тарифу*

*if maubytnya\_cina < teperishnuy\_rahynok:#Перевірка, чи наступний тариф вигідніший*

*return nastypnuy\_tarif#Повернення наступного тарифу, якщо він вигідніший*

*return None#Якщо наступний тариф не вигідніший*

**Лістинг коду 4.6(Main)**

*def main():*

*print("Програма для розрахунку вартості інтернет-тарифу.")*

*try:*

*tarif = int(input("Введіть тариф (1000, 2000 або 5000): "))*

*vukorustani\_mb = int(input("Введіть кількість витрачених мегабайтів: "))*

*vsya\_cina = pidrahynok\_cheky(tarif, vukorustani\_mb)*

*print(f"Загальна вартість: {vsya\_cina:.2f} грн.")*

*kraschuy\_tarif = kraschuy\_rahynok(tarif, vukorustani\_mb)#Перевірка, чи є вигідніший тариф*

*if kraschuy\_tarif:#Якщо вигідніший тариф існує, обчислюється його вартість і виводиться повідомлення*

*better\_cost = pidrahynok\_cheky(kraschuy\_tarif, vukorustani\_mb)*

*print(f"Якби ви перейшли на тариф {kraschuy\_tarif}, ви б заплатили {better\_cost:.2f} грн.")*

*except ValueError as e:*

*print(f"Помилка: {e}")*

*except Exception as e:*

*print(f"Сталася неочікувана помилка: {e}")*

*if \_\_name\_\_ == "\_\_main\_\_":*

*main()*

**3. Задачі з циклічним повторенням(**3, 8, 10, 15**)**

3.Виведіть на екран символи таблиці ASCII від 0 до 127 разом з їх номером у десятковій, двійковій, вісімковій та шістнадцятковій системах числення.

8. Оцінка студента за проходження курсу обчислюється як сума, 30% якої складають оцінки за практичні заняття (максимум 10 балів), а 70% - за екзамен (максимум 10 балів), якщо обидва бали перевищують або рівні 5. Інакше обирається мінімальна з отриманих. Напишіть програму, яка буде неперервно зчитувати пару оцінок (за практичні та екзамен), а потім виводитиме остаточну оцінку, поки не буде введено -1. Перед завершенням програма має вивести середню оцінку по групі.

10. Автентичність пристрою можна перевірити за допомогою отриманої з IMEI цифри Луна, порівнявши її з цифрою виробниками.Отримати цифру Луна можна таким чином:

1. Додати 14 цифр ІМЕІ так, що цифри на парних позиціях подвоюються при входженні в суму (якщо утворюється дворозрядне число, треба додати його розряди), а на непарних – залишаються з тим же значенням. Наприклад, для ІМЕІ 357683036257378
2. Якщо останній розряд суми дорівнює 0, це і є цифра Луна. Інакше потрібно відняти отримане значення від 10.  
   3+(2×5)+7+(2×6)+8+(2×3)+0+(2×3)+6+(2×2)+5+(2×7)+3+(2×7)

=3+(1+0)+7+(1+2)+8+(6)+0+6+6+(4)+5+(1+4)+3+(1+4)=**68**.Звідси, цифра Луна – 10-8=2. Перевірте введений з клавіатури ІМЕІ, який закінчується на 6, на відповідність цифрі Луна.

15.Напишіть програму, яка конвертує десяткове число в двійкове. Зчитуйте десяткове число від користувача як ціле число, а потім використовуйте алгоритм ділення для виконання перетворення.   
Ініціалізувати змінну result порожнім рядком

Оголосити змінну q – число для перетворення

**repeat**

Оголосити r та присвоїти йому значення остачі від ділення q на 2

Звести r до рядкового типу та додати її до початку

result

Поділити націло q на 2 та зберегти результат у q

**until q не доівнює 0**

Коли алгоритм завершує роботу, змінна result міститиме двійкове представлення числа. Відобразіть результат на екрані з відповідним пояснювальним текстом.

**Лістинг коду до задачі №3(Завдання 3)**

**Лістинг 1.1(функція яка виводить таблицю символів ASCII разом з їх кодами)**

*def print\_ascii\_tablucya():#виводить таблицю символів ASCII разом з їх кодами*

*print("Десяткова | Двійкова | Вісімкова | Шістнадцяткова | Символ")*

*print("-" \* 50)*

*for code in range(128): # Перебір кодів*

*char = chr(code) # Символ за кодом*

*binary = bin(code) # в 2ічну*

*octal = oct(code) # в 8мкову*

*hexadecimal = hex(code) # в 16кову*

*print(f"{code:^9} | {binary:^8} | {octal:^8} | {hexadecimal:^14} | {char:^6}")#Виведення рядка таблиці*

*if \_\_name\_\_ == "\_\_main\_\_":#запуск коду*

*print\_ascii\_tablucya()*

**Лістинг коду до задачі №8(Завдання 3)  
Лістинг 2.1(rozrahynok\_kincevoi\_ocinku)**

*def rozrahynok\_kincevoi\_ocinku(oсinka\_za\_praktuky, ocinka\_za\_ekzamen):# функції, яка приймає дві оцінки: за практичні заняття та екзамен*

*if oсinka\_za\_praktuky >= 5 and ocinka\_za\_ekzamen >= 5:#Перевірка, чи обидві оцінки більше або рівні 5*

*kinceva\_ocinka = 0.3 \* oсinka\_za\_praktuky + 0.7 \* ocinka\_za\_ekzamen#Якщо умова виконується, кінцева оцінка обчислюється як: 30% від оцінки за практичні,70% від оцінки за екзамен*

*else:#Якщо хоча б одна з оцінок менше 5*

*kinceva\_ocinka = min(oсinka\_za\_praktuky, ocinka\_za\_ekzamen)#Кінцева оцінка обирається як мінімальна з двох оцінок*

*return kinceva\_ocinka*

**Лістинг 2.1(main)**

def main():

ocinky = []#порожній список для зберігання оцінок

while True:

oсinka\_za\_praktuky = input("Оцінка за практичні заняття: ")#оцінка за практичну роботу від користувача

if oсinka\_za\_praktuky == "-1":

break#якщо ввести -1 цикл закінчиться

ocinka\_za\_ekzamen = input("Оцінка за екзамен: ")#оцінка за екзамен від користувача

try:

oсinka\_za\_praktuky = float(oсinka\_za\_praktuky)#Переведення оцінки за практичні на число з плаваючою точкою

ocinka\_za\_ekzamen = float(ocinka\_za\_ekzamen)#те саме тільки за екзамен

if not (0 <= oсinka\_za\_praktuky <= 10 and 0 <= ocinka\_za\_ekzamen <= 10):#Перевірка, чи оцінки знаходяться в межах від 0 до 10

print("Помилка: Оцінки повинні бути в межах від 0 - 10.")

continue#навіть якщо оцінки не в межах,то код працюватиме далі

kinceva\_ocinka = rozrahynok\_kincevoi\_ocinku(oсinka\_za\_praktuky, ocinka\_za\_ekzamen)#Обчислення кінцевої оцінки

ocinky.append(kinceva\_ocinka)#Додавання кінцевої оцінки до списку

print(f"Кінцева оцінка: {kinceva\_ocinka:.2f}")

except ValueError:

print("Помилка: Будь ласка, введіть числові значення.")

if ocinky:#перевірка чи порожній список

serednya\_ocinka = sum(ocinky) / len(ocinky)#Обчислення середньої оцінки

print(f"Середня оцінка по групі: {serednya\_ocinka:.2f}")

else:

print("Оцінки не були введені.")

if \_\_name\_\_ == "\_\_main\_\_":#Запуск коду

main()

**Лістинг коду до задачі №10(Завдання 3)  
Лістинг 3.1(rozrahynok\_chusla\_lyna)**

*def rozrahynok\_chusla\_lyna(imei):#приймає IMEI та обчислює цифру Луна*

*suma = 0 #буде зберігати суму IMEI*

*for i, chuslo in enumerate(imei):#перебирає кожну цифру IMEI разом з її індексом*

*chuslo = int(chuslo)#Перетворення символу цифри на ціле число*

*if i % 2 == 1:#Перевірка, чи індекс цифри є парним*

*chuslo \*= 2#Якщо індекс парний, цифра подвоюється*

*if chuslo > 9:#Перевірка, чи подвоєна цифра більше 9*

*chuslo = chuslo // 10 + chuslo % 10#Якщо так, цифра замінюється на суму її розрядів*

*suma += chuslo#Додавання обробленої цифри до загальної суми*

*chuslo\_luna = (10 - (suma % 10)) % 10#Обчислення цифри Луна*

*return chuslo\_luna*

**Лістинг 3.2(perevirka\_avtentychnosti)**

*def perevirka\_avtentychnosti(imei):#Перевірка автентичності ІMEI*

*if len(imei) != 15:#Перевірка, чи довжина IMEI = 15 цифрам*

*return False# Якщо ні,то False*

*imei\_bez\_luna = imei[:14]#Витягнення перших 14 цифр IMEI(бо 15та то число Луна)*

*chuslo\_luna = rozrahynok\_chusla\_lyna(imei\_bez\_luna)#Обчислення цифри Луна для перших 14 цифр IMEI*

*return chuslo\_luna == int(imei[-1])#Порівняння обчисленої цифри Луна з останньою цифрою IMEI. Якщо вони збігаються, функція повертає True, інакше — False.*

**Лістинг 3.3(main)**

*def main():*

*while True:*

*imei = input("Введіть IMEI (15 цифр, який закінчується на 6): ")#Зчитування IMEI від користувача*

*if imei.lower() == "вийти":#Перевірка, чи користувач ввів "вийти"*

*break*

*if len(imei) != 15 or not imei.endswith("6"):#Перевірка, чи IMEI містить 15 цифр і закінчується на 6*

*print("Помилка: IMEI має містити 15 цифр і закінчуватися на 6.")*

*continue*

*if not imei.isdigit():#Перевірка, чи IMEI містить лише цифри*

*print("Помилка: IMEI має містити лише цифри.")*

*continue*

*if perevirka\_avtentychnosti(imei):#Перевірка автентичності IMEI*

*print("IMEI автентичний.")*

*else:*

*print("IMEI не автентичний.")*

*if \_\_name\_\_ == "\_\_main\_\_": #Запуск програми*

*main()*

**Лістинг коду до задачі №15(Завдання 3)  
Лістинг 4.1(desyatkove\_do\_dviykovogo)**

*def desyatkove\_do\_dviykovogo(q):#приймає ціле десяткове число і повертає його у вигляді рядка.*

*result = " " #Ініціалізація порожнього рядка*

*while q > 0:#виконується, доки q більше 0*

*r = q % 2#Обчислення остачі*

*result = str(r) + result#правильний порядок цифр у двійковому представленні*

*q = q // 2#Цілочисельне ділення*

*return result if result else "0"#Повернення двійкового представлення. Якщо result порожній*

**Лістинг 4.2(golovna)**

*def golovna():*

*try:*

*q = int(input("Введіть десяткове число: "))*

*dviykovuy\_result = desyatkove\_do\_dviykovogo(q)*

*print(f"Двійкове представлення числа {q}: {dviykovuy\_result}")#Виведення результату у форматі: "Двійкове представлення числа X: Y"*

*except ValueError:*

*print("Помилка: Будь ласка, введіть ціле число.")*

*if \_\_name\_\_ == "\_\_main\_\_":#Запуск коду*

*golovna()*

**4.** [**Дизайн тестів**](https://nus-cs2103-ay1920s2.github.io/website/se-book-adapted/chapters/testCaseDesign.html)**. Еквівалентне розбиття та аналіз граничних значень**

Розглянемо операцію dispatch(Resource, Task) з системи управління екстреними викликами (дзвінки щодо пожеж, злочинів та ін.). Задача (task) може потребувати багато ресурсів різних типів. Наприклад, задача ‘пожежа в Clementi MRT’ може потребувати дві пожежні бригади та одну швидку допомогу.

* dispatch(Resource r, Task t):void
* Метод перерозподіляє ресурс r для задачі t. Оскільки відбувається передача лише одного ресурсу, його потрібно застосовувати кілька разів для задачі з багатьма ресурсами.

Розробіть тестові випадки, щоб відтестувати метод dispatch(Resource,Task), враховуючи еквівалентне розбиття та аналіз граничних значень. Які різні вхідні дані будуть комбінуватись для тестування методу?

(Тут не знала як зробити завдання,тому нижче таблиця(шось типу баг репорта),а в репозиторій закину вам реалізацію тест кейсів в директорії “Test\_ceysu”)

| ID | Опис багу | Кроки для відтворення | Очікуваний результат | Фактичний результат | Статус |
| --- | --- | --- | --- | --- | --- |
| 1 | Невідомий тип ресурсу не обробляється коректно | Викликати метод rozpodil з ресурсом типу "неіснуючий тип" і задачею "пожежа" | Помилка: невідомий тип ресурсу. | Помилка: невідомий тип ресурсу. | Пройдено |
| 2 | Невідома задача не обробляється коректно | Викликати метод rozpodil з ресурсом "пожежна бригада" і задачею "неіснуюча задача" | Помилка: невідома задача | Помилка: невідома задача | Пройдено |
| 3 | Задача, яка не потребує ресурсів, не обробляється коректно | Викликати метод rozpodil з ресурсом "пожежна бригада" і задачею "пожежа" (потребує []) | Помилка: задача не потребує ресурсів | Помилка: задача не потребує ресурсів | Пройдено |
| 4 | Недостатня кількість ресурсів не обробляється коректно | Викликати метод rozpodil з ресурсом "пожежна бригада" і задачею "пожежа" (потребує ["швидка допомога"]) | Помилка:  недостатня кількість ресурсів | Помилка:  недостатня кількість ресурсів | Пройдено |
| 5 | Ресурс не призначається для задачі, якщо він не відповідає вимогам задачі | Викликати метод rozpodil з ресурсом "пожежна бригада" і задачею "пожежа" (потребує ["швидка допомога"]) | Помилка:  недостатня кількість ресурсів | Помилка:  недостатня кількість ресурсів | Пройдено |
| 6 | Ресурс успішно призначається для задачі з максимальною кількістю ресурсів | Викликати метод rozpodil з ресурсом "пожежна бригада" і задачею "велика пожежа" (потребує ["пожежна бригада", "швидка допомога", "поліція"]) | Ресурс 'пожежна бригада' успішно призначено для задачі 'велика пожежа' | Ресурс 'пожежна бригада' успішно призначено для задачі 'велика пожежа' | Пройдено |
| 7 | Ресурс не призначається для задачі, якщо задача не потребує ресурсів | Викликати метод rozpodil з ресурсом "пожежна бригада" і задачею "пожежа" (потребує [“пожежна бригада”]) | Помилка: задача не потребує ресурсів | Помилка: задача не потребує ресурсів | Пройдено |
|  |  |  |  |  |  |

**5. Дизайн тестів. Таблиця прийняття рішень**

Побудуйте таблицю прийняття рішень та тестові сценарії

**Таблиця прийняття рішень**

| **Статус клієнта** | **Сума покупки** | **Базова**  **знижка** | **Додаткова знижка** | **Загальна знижка** |
| --- | --- | --- | --- | --- |
| Новий клієнт | <= 15000 | 5% | 0% | 5% |
| Новий клієнт | > 15000 | 5% | 5% | 10% |
| Постійний клієнт | <= 15000 | 10% | 0% | 10% |
| Постійний клієнт | > 15000 | 10% | 5% | 15% |
| VIP-клієнт | <= 15000 | 20% | 0% | 20% |
| VIP-клієнт | > 15000 | 20% | 5% | 25% |

**Тестові сценарії**

| **ID** | **Статус клієнта** | **Сума покупки** | **Очікувана знижка** | **Опис сценарію** |
| --- | --- | --- | --- | --- |
| 1 | Новий клієнт | 10000 | 5% | Новий клієнт із сумою покупки <= 15000 грн отримує базову знижку 5% |
| 2 | Новий клієнт | 20000 | 10% | Новий клієнт із сумою покупки > 15000 грн отримує базову знижку 5% +  додаткову 5% |
| 3 | Постійний клієнт | 10000 | 10% | Постійний клієнт із сумою покупки <= 15000 грн отримує базову знижку 10% |
| 4 | Постійний клієнт | 20000 | 15% | Постійний клієнт із сумою покупки > 15000 грн отримує базову знижку 10% + додаткову 5% |
| 5 | VIP-клієнт | 10000 | 20% | VIP-клієнт із сумою покупки <= 15000 грн отримує базову знижку 20% |
| 6 | VIP-клієнт | 20000 | 25% | VIP-клієнт із сумою покупки > 15000 грн отримує базову знижку 20% + додаткову 5% |

**Висновок:**

Виконавши практичну роботу

я навчилась виконувати різні задачі та правильно розуміти їх умову

**Автор програм:** Малишева Валерія Олексіївна  
**Час компіляції:** 01:38 (23/03/2025)