

**Практична робота №3**  
**Фундаментальні принципи ООП. Інкапсуляція**  
**Система оцінювання**

№	Тема	К-ть балів
1.	Завдання 1	0,9*
2.	Завдання 2	0,7*
3.	Завдання 3	1*
4.	Завдання 4	1*
5.	Здача звіту	0,4
	<b>Всього</b>	<b>4</b>

\* – діє бонусна система

**Завдання на практичне заняття**

1. **(Визначення класу).** Створіть клас BankAccount, який моделюватиме банківський рахунок користувача. Клас повинен включати приватні поля:

- id: int
- balance: decimal
- currency: Currencies (окреме перелічення з доступними валютами рахунку)

та публічні властивості:

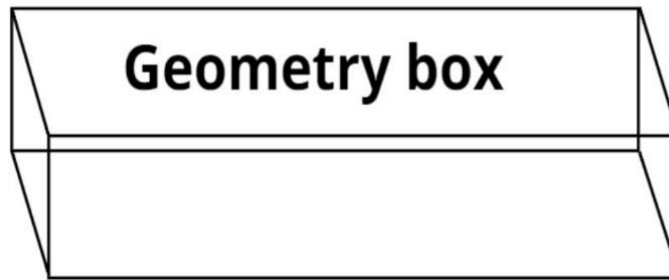
- Id: int
- Balance: decimal
- Currency: Currencies

Конструктори класу повинні передбачати автоматичне визначення номеру рахунку (+1 від попереднього номеру, починаючи з 1, без передачі відповідного параметру). За умовчанням баланс встановлюється нульовим, а валюта – гривня.

Передбачте методи:

- Deposit(decimal amount): void – вклад коштів на рахунок. Якщо сума перевищує 50000 грн, 2000\$ або €2000, слід вивести попередження, що інформація про дану транзакцію буде передана в податкову службу;
- Withdraw(decimal amount): void – зняття коштів з рахунку. Якщо сума коштів для зняття перевищує баланс рахунку, слід вивести попередження про нестачу коштів та заборонити зняття цієї суми.
- ShowInfo(): void – показ поточного стану рахунку.

2. **(Конструктор з валідацією)** Змодельуйте геометричну фігуру «бокс» (ящик) з параметрами «довжина», «ширина» та «висота». Розкрийте в інтерфейсі класу лише методи для знаходження площі поверхні, площі бокової (lateral) поверхні та об'єму ([формули](#)).



Кожна сторона боксу повинна бути невід'ємним числом, тому додайте перевірку параметрів, які передаються в конструктор. Створіть приватний сеттер, який перевірятиме параметри, замінюватиме від'ємні чи нульові значення на 1 та повідомлятиме про це в повідомленні для користувача.

3. **(Два класи)** Створіть класи Person (особа) та Product (товар). У кожної особи є ім'я, кошти та корзину товарів. Кожний товар має назву (непорожній рядок) та ціну (невід'ємне число). Розробіть програму, в якій кожна команда відповідає купівлі особою одного товару. Якщо особа може собі дозволити покупку товару, додайте цей товар у її корзину. Якщо грошей недостатньо, виведіть повідомлення ("[Ім'я особи] can't afford [Назва товару]").

Перші 2 рядки вводу повинні перелічувати всіх осіб з їх бюджетом та всі товари з їх цінами. Після всіх покупок виведіть для кожної особи в порядку появи всі придбані нею товари. Якщо нічого не придбано, виведіть ім'я особи та повідомлення "Nothing bought". Для від'ємної суми коштів особи слід викинути виняток з текстом "Money cannot be negative", а при порожньому рядку з іменем особи – "Name cannot be empty".

Ввід	Вивід
Pesho=11;Gosho=4 Bread=10;Milk=2; Pesho Bread Gosho Milk Gosho Milk Pesho Milk END	Pesho bought Bread Gosho bought Milk Gosho bought Milk Pesho can't afford Milk Pesho - Bread Gosho - Milk, Milk
Mimi=0 Kafence=2 Mimi Kafence END	Mimi can't afford Kafence Mimi – Nothing bought
Jeko=-3 Chushki=1; Jeko Chushki END	Money cannot be negative

4. **(Калорійність піци)** Піца готується з тіста та різних начинок. Змодельуйте клас *Pizza*, в якому передбачаються поля, що описують *назву піци*, *основу* та *набір начинок*. Кожний інгредієнт повинен мати свій клас з відповідними характеристиками:

- тісто може бути білим чи з цільнозернової муки, а також хрустким (crispy), липким (chewy) чи домашнім (homemade) тощо.
- начинки можуть включати м'ясні, овочеві, сирні чи соусні добавки.

Кожний інгредієнт описується масою в грамах, а також повинен мати метод для обчислення калорійності залежно від типу начинки. Калорійність на грам будемо обчислювати за допомогою спеціальних множників. Наприклад, біле тісто має множник 1.5, липке – 1.1, тобто 100г липкого білого тіста буде мати  $100 \cdot 1.5 \cdot 1.1 = 330$  калорій.

Змодельуйте класи таким чином, щоб вони були правильно інкапсульовані та забезпечували публічний метод, який обчислюватиме калорійність кожної піци відповідно до її інгредієнтів.

*Тісто.* Створіть клас, який представлятиме тісто для піци. Він включатиме

- *тип основи*, який може бути білим (множник – 1.5) або цільнозерновим (1.0);
- *метод приготування*, що дає хрустке (0.9), липке (1.1) чи домашнє (1.0) тісто;
- *масу в грамах*.

Клас повинен розкривати *лише геттер* для отримання калорій на грам. Встановіть правильні модифікатори доступу, конструктор, поля та геттери / сеттери.

Виконайте валідацію даних в сеттерах: якщо передано некоректне значення основи чи методу приготування, повинен викидатись виняток з повідомленням «Invalid type of dough». Маса тіста знаходиться в межах від 1 до 200 грамів включно – інакше трапиться виняток з повідомленням "Dough weight should be in the range [1..200].".

Протестуйте роботу класу з такими вводом та виводом:

Ввід	Вивід
Dough White Chewy 100 END	330.00
Dough Tip500 Chewy 100 END	Invalid type of dough.
Dough White Chewy 240 END	Dough weight should be in the range [1..200].

*Начинки (toppings).* Відповідний клас представлятиме 4 типи начинок: *м'ясо*, *овочі*, *сир* та *соус*. Кожна начинка має *масу* в грамах та множник для обчислення калорій:

- м'ясо – 1.2;
- овочі – 0.8;
- сир – 1.1;
- соус – 0.9.

Встановіть правильні модифікатори доступу, конструктор, поля та геттери / сеттери. Валідація даних передбачає передачу коректного типу начинки – інакше буде викинуто виняток з повідомленням "Cannot place [незва некоректного аргументу] on top of your pizza.". Дозволена маса начинки знаходиться в межах від 1 до 50 грам включно.

Інакше буде викинуто виняток з повідомленням "[Назва виду начинки] weight should be in the range [1..50].".

Протестуйте роботу класів так, щоб зчитувались одне тісто та одна начинка, а виводились їх калорійність:

Ввід	Вивід
Dough White Chewy 100 Topping meat 30 END	330.00 72.00
Dough White chewy 100 Topping Krenvirshi 500 END	330.00 Cannot place Krenvirshi on top of your pizza.
Dough White Chewy 100 Topping Meat 500 END	330.00 Meat weight should be in the range [1..50].

*Піца.* Повинна мати *назву* піци, *набір начинок* та *тісто*. Використайте раніше створені класи, а також додайте відкриті геттери для назви, кількості начинок та загальної кількості калорій (сума калорій окремих інгредієнтів). Розкрийте метод для додавання начинки, відкритий сеттер для тіста і геттер для загальної кількості калорій.

Валідація даних передбачає, що назва піци не повинна бути порожнім рядком, але й не довше 15 символів. Інакше повинен викидатись виняток з повідомленням "Pizza name should be between 1 and 15 symbols.". Кількість начинок повинна знаходитись у діапазоні від 1 до 10 включно. Інакше повинен викидатись виняток з повідомленням "Number of toppings should be in range [0..10].".

Протестуйте роботу програми на таких випадках:

Ввід	Вивід
Pizza Meatless Dough Wholegrain Crispy 100 Topping Veggies 50 Topping Cheese 50 END	Meatless - 370.00 Calories.
Pizza Burgas Dough White Homemade 200 Topping Meat 123 END	Meat weight should be in the range [1..50].
Pizza Bulgarian Dough White Chewy 100 Topping Sauce 20 Topping Cheese 50 Topping Cheese 40 Topping Meat 10 Topping Sauce 10	Number of toppings should be in range [0..10].

Topping Cheese 30 Topping Cheese 40 Topping Meat 20 Topping Sauce 30 Topping Cheese 25 Topping Cheese 40 Topping Meat 40 END	
Pizza Bulgarian Dough White Chewy 100 Topping Sirene 50 Topping Cheese 50 Topping Krenvirsh 20 Topping Meat 10 END	Cannot place Sirene on top of your pizza.

