Ćwiczenia

Krzysztof Gębicz

Ćwiczenie 1

Treść zadania:
Utwórz klasę `Prostokat` z `double szerokosc` i `double wysokosc`. Zaimplementuj konstruktor, który przyjmuje szerokość i wysokość. Dodaj metody:

- `public double pole()` — zwraca pole prostokąta,
- `public double obwod()` — zwraca obwód prostokąta.

Wymagania techniczne / dodatkowe warunki:

- Dodaj metodę `main`, która tworzy prostokąt i wypisuje pole i obwód.

Kroki testowe / przykłady:

- Stwórz `Prostokat p = new Prostokat(3, 4);`

Ćwiczenie 2

Treść zadania:
Utwórz klasę `Samochod` z prywatnymi polami: `String marka`, `String model`, `int rokProdukcji`, `int przebieg`. Zaimplementuj konstruktor ustawiający wszystkie pola. Dodaj:

- `public void jedz(int km)` — zwiększa przebieg o `km` (jeśli `km`>0),
- `public void setPrzebieg(int przebieg)` — ustawia przebieg, walidacja: przebieg nie może być ujemny,
- `public void przedstaw()` — wypisuje info o samochodzie w czytelnej formie.

Wymagania techniczne:

- Pola prywatne, setter z walidacją (rzuca `IllegalArgumentException` przy próbie ustawienia ujemnego przebiegu).
- `jedz(int km)` nic nie robi dla `km <= 0`.
- Zaimplementuj `main`, który pokaże działanie metod.

Kroki testowe / przykłady:

- Stwórz `Samochod s = new Samochod("Toyota", "Corolla", 2010, 120000)`, wywołaj `s.przedstaw()`, `s.jedz(150)`, `s.przedstaw()`.
- Spróbuj `s.jedz(-50)` — przebieg nie powinien się zmienić.
- Spróbuj ustawić przebieg na `-10` przez `setPrzebieg(-10)` — oczekuj wyjątku.

Ćwiczenie 3

Treść zadania:
Utwórz klasę Uczen z polami String imie, String nazwisko i ArrayList<Integer> oceny. Zaimplementuj:

- konstruktor ustawiający imię i nazwisko,
- public void dodajOcene(int ocena) — dodaje ocenę (tylko 1..6),
- public double srednia() — zwraca średnią ocen (0.0 jeśli brak ocen),
- public void przedstawSie() — wypisuje imię + nazwisko + średnia.

Wymagania techniczne:

- Użyj ArrayList<Integer> do przechowywania ocen.
- Waliduj oceny w dodajOcene (1..6).
- srednia() powinna zwracać double z obliczoną średnią.

Kroki testowe / przykłady:

- Dodaj oceny 5,4,3 — średnia powinna być 4.0.
- Dla ucznia bez ocen srednia() zwraca 0.0.

Ćwiczenie 4

Treść zadania:
Utwórz klasę `BankAccount` z prywatnymi polami: `String numerKonta` i `double saldo`. Dodaj pole statyczne `private static int liczbaKont`, które zlicza utworzone instancje. Zaimplementuj:

- konstruktor `BankAccount(String numerKonta, double saldoPoczatkowe)` — zwiększa `liczbaKont`,
- `public void wplac(double kwota)` — dodaje środki (tylko gdy `kwota > 0`),
- `public void wyplac(double kwota)` — odejmuje środki (tylko gdy `kwota > 0 && kwota <= saldo`),
- `public double getSaldo()` oraz `public static int getLiczbaKont()`.

Wymagania techniczne:

- Walidacja przy depozycie i wypłacie (nie przyjmować ujemnych kwot).
- `liczbaKont` jest `static` i zwraca całkowitą liczbę stworzonych kont.

Kroki testowe / przykłady:

- Utwórz dwa konta — sprawdź `BankAccount.getLiczbaKont()` równe 2.
- Wpłać i wypłać kwoty, spróbuj wypłacić więcej niż saldo — saldo nie powinno się obniżyć poniżej zera.