

#### Laborator 4. Obiecte

1. Să se scrie o clasă Cerc, care are un constructor în care indicați raza cercului, împreună cu metodele `getArie()` pentru a obține aria și `getCircumferinta()`, care returnează circumferința cercului. Să se testeze funcționalitatea acestei clase.
2. Să se implementeze o clasă Produs. Fiecare produs are o denumire și un preț unitar. Scrieți constructorul și următoarele metode: `afiseazaProdus()`, `getPret()`, `setPret(int np)`, `aplicaReducere(int procent)`. Să se scrie un program care creează două produse și le afișează, apoi să li se reducă prețul cu 10% și apoi să se afișeze din nou.
3. Creați o clasă ContBancar care are două variabile instanță private numite `numarCont` și `sumaCont`. Aceste variabile sunt inițializate prin intermediul parametrilor unui constructor definit de către programator. Să se implementeze metode publice pentru adăugarea, respectiv extragerea, unei sume în/din cont. Apoi să se scrie o altă clasă Test care, în metoda `main`, creează un obiect de tip ContBancar cu numărul 40 și valoarea minimă 10 RON și apoi adaugă, respectiv scoate o sumă din cont. După fiecare apelare a unei metode să se afișeze numărul contului și soldul curent.
4. Să se creeze o clasă denumită Dreptunghi. Clasa are atributele lungime și latime, fiecare inițializate implicit cu 1, și metode care calculează perimetrul și aria dreptunghiului. În plus, clasa conține metode `set` și `get` pentru ambele atribute. Metodele `set` ar trebui să verifice faptul că lungimea și lățimea au valori reale între 0.0 și 20.0. Să se scrie o altă clasă care verifică funcționalitatea clasei Dreptunghi.
5. Să se creeze o clasă denumită Dreptunghi care este puțin mai complicată decât cea din problema anterioară. Această clasă stochează numai coordonatele colțurilor dreptunghiului. Constructorul apelează o metodă `set()` care acceptă patru seturi de coordonate și verifică dacă acestea nu au coordonatele  $x$  și  $y$  mai mari de 20.0. De asemenea, verifică faptul că cele 4 seturi de coordonate specifică un dreptunghi. Metodele acestei clase calculează lungimea, lățimea, perimetrul și aria fiecărui dreptunghi. Includeți o metodă booleană `estePatrat()` care determină dacă dreptunghiul este pătrat. Să se scrie o altă clasă care verifică funcționalitatea acestei clase.
6. Să se scrie o clasă Complex care realizează operații aritmetice cu numere complexe. Numerele complexe au forma:  $\text{parteaReala} + i \cdot \text{parteaImaginara}$ , unde  $i = \sqrt{-1}$ . Folosiți variabile double pentru a memora datele private ale clasei. Furnizați 2 constructori, unul care inițializează variabilele clasei și un constructor care le inițializează cu valori implicite. Scrieți metode publice pentru fiecare din următoarele operații:
  - a) Adunarea a două numere complexe.
  - b) Scaderea a două numere complexe. Rezultatul este un număr complex nou.
  - c) Afișarea unui număr complex sub forma  $(a,b)$  unde  $a$  este partea reală, iar  $b$  partea imaginară a numărului complex.

Să se scrie o altă clasă care testează funcționalitatea clasei Complex.

7. Creați o clasă numită **Rational** pentru realizarea operațiilor aritmetice cu numere rationale. Folosiți variabile întregi pentru a reprezenta variabilele instanță private ale clasei – **numarator** și **numitor**. Scrieți un constructor care permite inițializarea obiectului când este creat, cu valorile primite ca parametri. Constructorul stochează fracția în forma redusă (de exemplu, fracția  $\frac{2}{4}$  ar trebui stocată în obiect ca  $\frac{1}{2}$ ). Scrieți un alt constructor care nu are parametri și care inițializează obiectul cu valori implicite.

Clasa Rational are următoarele metode publice:

- adunarea a două numere rationale. Rezultatul adunării ar trebui stocat în forma redusă.
- scaderea a două numere rationale.
- înmulțirea a două numere rationale.
- împartirea a două numere rationale.
- afișarea numerelor rationale în forma **a/b**
- afișarea numerelor rationale în format virgula flotantă.

Rezultatul efectuării unei operații va fi un număr rational nou, deci va trebui stocat în forma redusă într-un alt obiect de tip Rational.