

### Bilet 1

Se se realizeze un program care descrie organizarea personalului unei instituții folosind clase derivate. Clasa de baza este `Angajat`, care va retine numele, calificarea si salariul. In institutie exista mai multe clase de angajati. De exemplu: un șef de secție (`SefSectie`) este un angajat pentru care avem specificata numele sectiei pe care o conduce, `Director` care are indemnizatie de conducere. Construiti o lista a personalului.

### Bilet 2

Sa se defineasca clasa abstracta `Stiva` care dispune de 3 metode virtuale pure `push`, `pop` pentru punerea, scoaterea unui element din stiva si `isEmpty` pentru a testa daca stiva e vida. Sa se implementeze clasa `StivaInlantuta` care va reprezenta o stiva ca o lista simplu inlantuita.

### Bilet 3

Sa se defineasca clasa abstracta `Coda` care dispune de 3 metode virtuale pure `add`, `get` pentru adaugarea, scoaterea unui element din coada si `isEmpty` pentru a testa daca coada e vida. Sa se implementeze clasa `CodaInlantuta` care va reprezenta o stiva ca o lista inlantuita.

### Bilet 4

Sa se implementeze clasa `Judet` care contine o lista de localitati. Localitatile se impart in mai multe categorii: sate si orase. Orice localitate are un nume si un numar de locuitori. Un sat este o localitate. Un oras este o localitate care are un numar de blocuri.

### Bilet 5

Sa se implementeze o clasa care sa retina jurnalele si cartile dintr-o biblioteca. Orice articol dintr-o biblioteca are un titlu si o cota (codul de identificare a cartii in bibiloteca). Un jurnal este un articol din biblioteca care este caracterizat in plus de numar si ISSN (sir de caractere ce identifica unic jurnalul). O carte este un articol din biblioteca care are in plus urmatoarele atribute: autor, an aparitie, editura si ISBN (sir de caractere ce identifica unic cartea).

### Bilet 6

Se considera clasa `Abonat` derivata in clasa `Persoana`. Supraincarcati operatorii `<<`, `>>`. Sa se construiasca clasa `Agenda` ce contina o lista de abonati si sa se supraincarce operatorul `[]` (indexare) care returneaza abonatul cu numele precizat .

### Bilet 7

Implementati ierarhia de clase `Telefon`, `TelefonFix`, `TelefonMobil`. Orice telefon are asociat un numar. Un telefon fix poate fi cu fir sau fara fir. Un telefon mobil are o autonomie a bateriei si poate fi smart sau nu.

### Bilet 8

Sa se implementeze urmatoarea ierahie de clase: `Animal( nume)`, `Mamifer( nume, perioadaGestatie)`, `AnimalZburator( nume, altitudineZbor)`, `Liliac( nume, perioadaGestatie, altitudineZbor)`.

### **Bilet 9**

Sa se implementeze clasele `Cerc` si `Triunghi` derivate din `FiguraGeometrica`. Clasa `FiguraGeometrica` va fi definita ca si clasa abstractă având declarate metodele `getPerimetru()` si `getAria()` ca funcții virtuale pure.

### **Bilet 10**

Implementati o ierarhie de clase ce modeleaza o `Galaxie` ce este formata dintr-o multime de obiecte cosmice. Orice obiect cosmic este caracterizat prin nume, raza si o masa. Obiectele cosmice sunt de doua tipuri `Planete` si `Stele`. O planeta are o perioada de rotatie in timp ce o stea are o anumita stralucire.

### **Bilet 11**

Implementați clasa `Cantar`. Un cântar are o capacitate maximă admisă. Dacă se încearcă cântărirea unui obiect ce depășește cu maxim 10% greutatea maximă admisă se va genera excepția `AvertismentDepasireGreutate` iar dacă se depășește și această limită se va genera eroarea `DepasireGreutate`

### **Bilet 12**

Implementați clasa `Coadă` implementată sub forma unui tablou alocat dinamic. Gestionați excepțiile ce pot să apară.

### Bilet 13

Implementați clasa `Stiva` implementată sub forma unui tablou alocat dinamic. Gestionați excepțiile ce pot să apară.

### Bilet 14

Să se implementeze clasa `Student(nume, specializarea, an)` care deriva din clasa `Persoana(nume, varsta)`. Numele și specializarea se vor reține ca date membru ce se vor aloca dinamic. Să se implementeze următoarele metode:

- constructori de initializare și copiere
- funcții pentru supraincercarea operatorului de atribuire
- funcții pentru supraincercarea operatorilor `<<` și `>>` pentru afișarea și citirea datelor.

### Bilet 15

Să se implementeze ierarhia de clase `Masina (nume, marca)`, `MasinaElectrica(nume, marca, autonomie)`, `MasinaCuCombustie(nume, marca, tipCarburant, consum)`. Presupunând că `numele`, `marca` și `tipCarburant` sunt șiruri de caractere ce se vor aloca dinamic să se implementeze următoarele metode:

- constructori de initializare și copiere
- funcții pentru supraincercarea operatorului de atribuire
- funcții virtuale pentru afișarea și citirea datelor.

### Bilet 16

La o competiție participă o mulțime de concurenți ce pot fi elevi și studenți. Realizați o mulțime de clase care să permită gestiunea listei de concurenți.