



ACADEMIA DE SOFTWARE

MANAGEMENT ATELIER AUTO



Detaliile de implementare și evaluare sunt la finalul acestui document.

Se va realiza o aplicație pentru gestionarea unui atelier auto. Utilizatorul va avea la dispoziție un meniu afișat în consolă, prin intermediul căruia se vor realiza operațiunile legate de personalul atelierului și de mașini, astfel:

I. Angajați

Există 3 tipuri de angajați: Director, Mecanic și Asistent. Toți angajații au în comun câmpurile ID, Nume, Prenume, Data Nașterii, Data Angajării și Coeficient salariat.

Se vor implementa următoarele funcționalități: Afișare angajați, Adăugare angajat, Ștergere angajat, editare angajat și calcularea salariului unui angajat.

Observații:

- Id-ul trebuie să se incrementeze automat pentru fiecare angajat adăugat.
- Coeficientul salarial nu se setează de la tastatură. (Directorul are coeficientul 2, angajatul 1.5 și asistentul 1).
- Numele și prenumele nu trebuie să fie nule și nu trebuie să depășească 30 de caractere fiecare.
- Angajatul trebuie să aibă peste 18 ani, împliniți inclusiv în ziua angajării.
- Data angajării nu poate fi o dată din viitor.
- Ștergerea, editarea și calcularea salariului unui angajat se fac în baza ID-ului acestuia, introdus de la tastatură. Dacă ID-ul nu corespunde niciunui angajat, se afișează un mesaj corespunzător și se revine la introducerea ID-ului.
- Calculul salariului se face după formula **Ani vechime * coeficient * 1000 lei**.

II. Mașini

Există 3 tipuri de mașini: Standard, Autobuz și Camion. Toate mașinile au în comun câmpurile ID, Numărul de kilometri, Anul fabricației și un indicator dacă este motor Diesel sau nu. Fiecare tip de mașină în parte are următoarele specificații:

1. Mașinile standard au, în plus (față de câmpurile menționate general pentru toate mașinile), un mod de transmisie, care poate fi doar Manual sau Automat.
2. Autobuzele au, în plus (față de câmpurile menționate general pentru toate mașinile) un câmp care să indice Numărul de locuri.
3. Camioanele au, în plus (față de câmpurile menționate general pentru toate mașinile) un câmp care să indice tonajul.

Fiecare instanță a unei mașini trebuie să expună un mod de a calcula valoarea poliței de asigurare. Pentru aceasta, se țin cont de următoarele informații:

1. Pentru mașinile standard, polița de asigurare se calculează astfel:
 - Este vechimea (exprimată în ani) înmulțită cu 100, la care se adaugă următoarele:

- 500 dacă este motor Diesel;

- 500 dacă a depășit 200 000 de kilometri.

2. Pentru autobuze, polița de asigurare se calculează astfel:

- Este vechimea (exprimată în ani) înmulțită cu 200, la care se adaugă următoarele:

- 1000 dacă este motor Diesel;

- 1000 dacă au fost depășiți 200 000 de kilometri, 500 dacă au fost depășiți doar 100 000 de kilometri.

3. Pentru camioane, polița de asigurare se calculează astfel:

- Este vechimea (exprimată în ani) înmulțită cu 300, la care se adaugă 700 dacă au fost depășiți 800 000 de kilometri.

Formula poate fi apelată standard, sau în modul DISCOUNT, care înseamnă: o reducere de 5% pentru mașinile standard, de 10% pentru autobuze și de 15% pentru camioane.

III. Activitatea Atelierului

Într-un atelier pot lucra mai mulți angajați, care pot repara mai multe tipuri de mașini, având următoarele restricții:

1. Trebuie să existe cel puțin un angajat pentru ca atelierul să poată fi considerat deschis.

2. Angajații nu pot fi supraaglomerați: ei pot lua în primire, în total, cel mult 3 mașini normale și cel mult câte o mașină dintre celelalte două tipuri menționate anterior (maxim 1 autobuz, maxim 1 camion).

3. Pentru a putea gestiona acest lucru, angajații oferă un estimat de timp pentru a repara o mașină, și se asumă faptul că estimatul este corect.

4. Pe măsură ce un angajat finalizează repararea unei mașini, poate lua în primire și alte mașini, respectând restricțiile de mai sus.

5. Un nou vehicul venit pentru reparație poate să fie alocat primului angajat disponibil, sau poate solicita să aștepte strict intrarea la un anumit angajat. Dacă angajatul solicitat nu este disponibil, poate alege să plece sau să treacă la primul angajat liber.

6. Când toți angajații sunt complet ocupați, un vehicul poate fi pus în așteptare, sau proprietarul poate decide să plece.

Respectând acest scenariu, permiteți adăugarea de solicitări într-un meniu afișat în consolă, cât și mesaje specifice punctul în care ați ajuns: nu mai sunt locuri libere la un anumit angajat dorit, nu mai sunt locuri libere în atelier la niciun angajat etc. Afișați, în cadrul fiecărei operații, coada de așteptare.

IV. Raportare (opțional)

Pentru evaluarea activității atelierului, permiteți afișarea următoarelor rapoarte, selectate dintr-un meniu afișat în consolă:

1. Angajatul cu cea mai mare încărcare în data curentă.

2. Top 3 angajați care au reparat mașini ce însumează valoarea polițelor de asigurare maximă.

3. Top 3 angajați care au reparat cele mai multe autobuze noi (care au o vechime de mai puțin de 5 ani).
4. Numele și prenumele celor mai solicitați angajați (cei pentru care se fac cereri speciale de către deținătorii de vehicule, pentru a fi reparată mașina special de către aceștia).
5. Bacșișul fiecărui angajat, știind că primește 1% din valoarea poliței de asigurare cu DISCOUNT pentru fiecare mașină reparată.

Alte observații:

1. Se pot folosi și fișiere text pentru citirea/salvarea datelor, însă este suficient ca acestea să persiste în memorie.

Ce trebuie trimis:

1. Proiectul va fi dezvoltat în unul dintre limbajele C#, C++ sau Java. Este permisă și utilizarea Javascript, în condițiile în care se respecta paradigma OOP.
2. Trebuie să primim o arhivă care să conțină codul sursă, dezvoltat pe baza cerințelor anterioare. Aceasta va conține și executabilele, alături de eventuale instrucțiuni de rulare acolo unde este cazul.

Ce așteptăm:

1. Proiectul trebuie să fie dezvoltat respectând paradigma programării orientate pe obiect și să valorifice principiile esențiale OOP, cu accent pe moștenire și suprascriere.
2. Vrem să vedem că sunt cunoscute:
 - Manipularea generală de date (operații I/O, eventual și pe fișiere);
 - Lucrul cu clase (și clase abstracte);
 - Moșteniri;
 - Suprascrieri de funcții;
 - Capacitatea de înțelegere a anumitor cerințe și identificarea contextului în care acestea trebuie rezolvate.
3. Codul trebuie să fie coerent și lizibil, de preferat self-explanatory: numele claselor, variabilelor să fie intuitiv, iar comentariile să fie puse doar acolo unde este strict necesar.

Modalitate de evaluare:

Corectitudinea implementării (respectarea cerințelor și implementarea corectă și completă a acestora) și validarea datelor:

a. Partea I – Angajat: (34p)

- Crearea clasei de bază Angajat: 3p.
- Crearea angajaților specializați: 2p / tip de angajat.
- Fiecare acțiune (adăugare, afișare, ștergere, editare, calculare salariu): 3p / acțiune.
- Setare ID auto-increment 2p.
- Setare Coeficient salarial automat 2p.
- Validări (vârstă, ID la acțiuni): 4p + 2p.

b. Partea a II-a – Mașină: (27p)

- Crearea clasei de bază Mașină: 3p.
- Crearea mașinilor specializare: 2p / tip de mașină.
- Implementarea formulelor de calcul pentru asigurare: 4p / formulă pentru fiecare clasă.
- (+) Implementarea formulelor ținând cont de DISCOUNT: 2p / formulă.

c. Partea a III-a – Atelier: (39p)

- Asigurarea faptului că atelierul poate fi considerat deschis dacă are cel puțin un angajat: 2p.
- Asigurarea restricțiilor din atelier pentru aglomerarea unui angajat: 3p.
- Implementarea corectă a cozii de așteptare la atelier: sunt tratate toate cazurile – este cerut un anumit angajat și acela este liber, este cerut un anumit angajat, și acela este ocupat (se poate aștepta, se poate cere trecerea la un alt angajat, liber, sau se poate pleca), există cel puțin un angajat liber și se poate rezolva problema, nu există niciun angajat liber și se poate aștepta sau se poate pleca. Prin angajat liber înțelegem angajat liber în raport cu tipul de mașină venit în solicitare. Sunt 7 cazuri de bază, fiecare valorează 4p.
- Coerența implementării pentru întreaga parte, validarea inputurilor: 6p.

Cele 3 părți însumează **100p**. Suplimentar, în limita timpului rămas, puteți aborda și Partea a IV-a, care va fi punctată **dacă și numai dacă** se obțin pe Părțile I și II minim 24p, respectiv 21p. Pentru partea a III-a, se cer minim 29p.

d. Partea a IV-a: (20p)

- Cerința 1: 2p.
- Cerința 2: 4p.
- Cerința 3: 4p.
- Cerința 4: 5p.
- Cerința 5: 5p.

Atenție! NU sunt luate în calcul sursele care nu respectă paradigma OOP!

În plus, se consideră bine știute citirea și afișarea datelor. Aplicația dezvoltată trebuie să fie ușor de utilizat, astfel că **se scad** din punctajul total 10p dacă este necesar să ne uităm în cod pentru a vedea ce trebuie să introducem (lipsește mesaje necesare astfel încât utilizarea să fie intuitivă) și **se acordă un bonus** de 5p dacă aplicația este intuitiv de utilizat iar datele sunt afișate ordonat, se curăță consola după fiecare utilizare și alte aspecte pe care le considerați importante din punctul de vedere al experienței utilizatorilor acestui proiect.

Timp de lucru: Din momentul primirii cerințelor pe mail, timpul de lucru alocat este de 2 zile.