**Lucrările de laborator la disciplina „Programarea Calculatoarelor” pentru anul de studii 2019-20!**

**Ultima actualizare: 25.09.2019**

**Notă:** pentru grupa de la masterat sunt obligatorii primele 10 lucrări de laborator, a 11-a și a 12-a sunt opționale. Numărul variantei este scris pe pagina următoare. Dacă este efectuată o altă variantă (ca excepție, la lucrarea nr. 1 se acceptă variantă atribuită inițial fiecărui student), lucrarea de laborator nu este acceptată. Dacă este indicat numele sau prenumele altui student sau dacă sunt dovezi clare că lucrarea este luată de la alt coleg, atunci aceasta va fi apreciată cel mult cu nota cinci (dacă studentul va explica amănunțit procesul efectuării ei se poate pune, ca excepție, nota șase).

Dacă studentul efectuează lucrarea de laborator în timpul perechii el este eliberat de necesitatea elaborării raportului, este suficient să transmită codul programului și câteva capturi de ecran pe adresa de e-mail [vmititelu.for.students@gmail.com](mailto:vmititelu.for.students@gmail.com). Studentul care nu au efectuat lucrarea de laborator pe parcursul perechii este obligat să elaboreze raportul, care trebuie să conțină un volum de circa 6-10 pagini, dintre care prima pagină trebuie să reprezinte foia de titlu (denumirile ministerului, universității, și facultății, denumirea disciplinei, numărul lucrării de laborator, tema, numele și prenumele studentului și a profesorului, loc pentru data prezentării, pentru notă și pentru semnătura profesorului). Pagina a doua începe cu scopul și sarcina lucrării, apoi trebuie să urmeze un scurt material teoretic la temă (circa o pagină), iar celelalte pagini trebuie să conțină codul programului (în întregime), precum și screen-shot-urile (minim trei, cu diverse valori inițiale, îndeosebi cele ce pot genera erori sau situații imprevizibile) care confirmă efectuarea lucrării de laborator respective și varianta electronică. Ultima pagină trebuie să conțină concluziile (minim trei) făcute pe baza îndeplinirii lucrării de laborator respective. Pentru a obține nota maximă studentul trebuie să prezinte lucrarea de laborator în timp rezonabil și trebuie să poată demonstra că lucrarea a fost îndeplinită de el (să răspundă la întrebările suplimentare referitor la lucrare). La sfârșit verificați obligatoriu ortografia și gramatica. Lucrările fără diacritice sau cu greșeli numeroase nu se acceptă.

**Variantele la lucrările de laborator:**

1. Andrieș Patricia-Georgiana
2. Baltean Alina
3. Beher Vladislav
4. Boboc Sorin
5. Bondari Cristian-Valeriu
6. Borzin Denis
7. Borzin Dorin
8. Botnariuc Dumitru
9. Bulgaru Ion
10. Corcenco Andrei
11. Feodorov Petru
12. Ghinda Vadim
13. Goga Cristin
14. Mangâr Catalina
15. Mîrzac Ion
16. Pîrgari Vadim
17. Pîrlea Ana-Maria
18. Pogor Carolina
19. Rotari Daniel
20. Timis Teodor
21. Stratulat Constantin
22. Rusu Victor

**Lucrarea de Laborator nr. 1**

**Tema: Instrucțiuni liniare și condiționale**

**Scopul:** scrierea programelor simple utilizând instrucțiuni liniare (de atribuire, citire și afișare a informației) și condiționale (if, switch- case) în limbajul C.

**Sarcini**

Scrieți un program care citește de la tastatură valoarea variabilei x, calculează valoarea expresiei indicate în variantă, apoi afișează pe ecran rezultatul (variabila y). Verificați cazurile care pot genera eroare (împărțire la zero, radical sau logaritm din număr negativ ș.a.). Demonstrați, din punct de vedere matematic, că rezultatul oferit de program este cel corect.

* 1. .
  2. .
  3. .
  4. .
  5. .
  6. .
  7. .
  8. .
  9. .
  10. .
  11. .
  12. .
  13. .
  14. .
  15. .
  16. .
  17. .

**Lucrarea de Laborator nr. 2**

**Tema: Instrucțiuni repetitive (cicluri)**

**Scopul:** utilizarea instrucțiunilor repetitive (for, while și do-while) pentru scrierea programelor în limbajul de programare C.

**Sarcini**

Scrieți un program care efectuează ceea ce este indicat în variantă, apoi afișează pe ecran rezultatul (se recomandă ca variantele 1-7 să fie efectuate cu instrucțiunea „for”, variantele 8-14 – cu „while”, iar variantele 15-22 – cu „do- while”):

1. calculează suma numerelor pare de la 2 la n
2. calculează n! (n factorial)
3. calculează suma șirului: 1+1/2+1/3+1/4+...+1/n
4. calculează produsul numerelor impare de la 1 la n
5. calculează suma șirului: 1-1/2+1/4-1/8+...+(-1)n/2n
6. determină dacă un număr este perfect (este egal cu suma tuturor divizorilor săi, în afară de însăși numărul)
7. determină dacă două numere sunt prietene (dacă fiecare din ele este egal cu suma divizorilor celuilalt număr)
8. calculează suma cifrelor unui număr
9. calculează câte cifre are un număr
10. determină dacă un număr este prim (se divide numai la 1 și la el însuși)
11. determină dacă 2 numere sunt prime între ele (cel mai mare divizor comun al lor este 1)
12. determină al n-lea număr Fibbonacci
13. determină cel mai mare divizor comun al două numere
14. determină toate numerele prime din intervalul [a;b]
15. citește de la tastatură numere până când introducem un număr negativ și determină suma și produsul numerelor introduse
16. citește de la tastatură numere până când introducem 0 și determină suma numerelor introduse
17. citește de la tastatură numere până când introducem un număr pozitiv și determină câte numere am introdus (numărul de numere)
18. citește de la tastatură numere până când introducem 0 și determină media aritmetică a numerelor introduse
19. citește de la tastatură numere până când introducem un număr pozitiv și determină câte numere am introdus (numărul de numere)
20. citește de la tastatură numere până când introducem 0 și determină câte numere pozitive și câte negative au fost introduse
21. citește de la tastatură numere până când introducem un număr negativ și determină media aritmetică a numerelor introduse
22. citește de la tastatură numere până când introducem 0 și determină suma numerelor pozitive și produsul numerelor negative care au fost introduse

**Lucrarea de Laborator nr. 3**

**Tema: Funcții care întorc valori**

**Scopul:** utilizarea funcțiilor pentru scrierea programelor în limbajul de programare C.

**Sarcini**

Scrieți un program care efectuează ceea ce este indicat în variantă, utilizând funcții, apoi afișează pe ecran rezultatul (utilizarea cel puțin a unei funcții care întoarce valori, declarată de voi, este obligatorie):

1. calculează media aritmetică a 3 numere întregi
2. calculează media geometrică a 2 numere reale
3. calculează suma cifrelor unui număr
4. calculează diagonala unui pătrat cu latura a
5. calculează aria unui dreptunghi cu laturile a și b
6. calculează distanța dintre 2 puncte:
7. calculează aria unui triunghi cu laturile a, b și c utilizând formula lui Heron:
8. calculează n! (n factorial)
9. calculează una din laturile triunghiului conform teoremei cosinusului: c2=a2+b2-2abcosC
10. calculează una din laturile triunghiului dreptunghic conform teoremei lui Pitagora: c2=a2+b2
11. rezolvă o ecuație de gradul I: ax+b=0
12. calculează forța lui Coulomb conform formulei:
13. calculează aria unui triunghi conform formulei:
14. calculează aria unui trapez:
15. calculează ab
16. calculează aria unui triunghi conform formulei , unde C este mărimea unghiului (în grade), iar a și b sunt lungimile laturile
17. determină dacă trei numere pot fi laturi ale unui triunghi (dacă suma lungimilor oricărei două laturi este mai mare decât lungimea celei de-a treia).
18. rezolvă o ecuație de gradul II: ax2+bx+c=0
19. calculează media armonică a 3 numere întregi
20. calculează câte cifre are un număr
21. calculează suma șirului: 1+1/2+1/3+1/4+...+1/n
22. calculează suma numerelor de la 1 la n

**Lucrarea de Laborator nr. 4**

**Tema: Vectori și matrice (tablouri)**

**Scopul:** utilizarea vectorilor și matricelor pentru scrierea programelor în limbajul de programare C.

**Sarcini**

Scrieți un program care prelucrează tablourile așa cum este indicat în variantă, apoi afișează pe ecran rezultatul utilizând funcția „printf”. Numărul de elemente ale tabloului și însăși elementele trebuie citite de la tastatură, utilizând funcția „scanf”. În probleme cu matrice ele se consideră cu număr diferit de rânduri și coloane, doar dacă este vorba despre diagonala principală sau secundară matricele sunt pătratice.

1. calculează media aritmetică a elementelor unui vector
2. calculează media aritmetică a elementelor unei matrice
3. calculează suma elementelor pare ale unui vector
4. calculează suma elementelor impare ale unei matrice
5. calculează produsul elementelor impare ale unui vector
6. calculează produsul elementelor pare ale unei matrice
7. determină câte elemente pare conține un vector
8. determină câte elemente divizibile cu 3 are un vector
9. calculează suma elementelor diagonalei principale ale unei matrice
10. calculează produsul elementelor diagonalei secundare ale unei matrice
11. determină câte elemente nenegative conține un vector
12. determină câte elemente pozitive conține o matrice
13. determină câte elemente divizibile cu 4 are o matrice
14. calculează suma elementelor de pe poziții pare într-un vector
15. calculează produsul elementelor de pe poziții impare într-un vector
16. determină elementul maxim din fiecare rând al matricei
17. sortează crescător un vector
18. determină elementul minim din fiecare coloană a matricei
19. sortează crescător elementele fiecărui rând al matricei
20. sortează descrescător elementele fiecărei coloane a matricei
21. calculează suma elementelor pozitive ale diagonalei principale a matricei
22. calculează produsul elementelor negative ale diagonalei secundare a unei matrice

**Lucrarea de Laborator nr. 5**

**Tema: Șiruri de caractere**

**Scopul:** utilizarea șirurilor de caractere pentru scrierea programelor în limbajul de programare C.

**Sarcini**

Scrieți un program care prelucrează șirurile de caractere așa cum este indicat în variantă, apoi afișează pe ecran rezultatul (în programele în care se cere de determinat dacă o careva literă este prezentă în text trebuie de optimizat codul):

1. determină dacă litera „a” se conține într-un șir de caractere
2. determină de câte ori litera „e” se conține într-un șir de caractere
3. inversează un șir de caractere
4. înlocuiește litera „a” cu litera „o” într-un șir de caractere
5. determină dacă litera „m” se conține într-un șir de caractere
6. determină de câte ori litera „n” se conține într-un șir de caractere
7. înlocuiește litera „i” cu litera „a” într-un șir de caractere
8. determină dacă cuvântul „info” se conține într-un șir de caractere
9. înlocuiește cuvântul „ori” cu cuvântul „sau” într-un șir de caractere
10. numără câte cuvinte sunt într-un șir de caractere
11. determină dacă cuvântul „programare” se conține într-un șir de caractere
12. înlocuiește cuvântul „sa” cu cuvântul „s-a” într-un șir de caractere
13. înlocuiește semnul „!” cu semnul „.” într-un șir de caractere
14. determină dacă litera „p” se conține într-un șir de caractere
15. determină de câte ori cuvântul „ASEM” se conține într-un șir de caractere
16. determină de câte ori se repetă fiecare cuvânt într-o propoziție
17. determină numărul de vocale într-o propoziție
18. determină numărul de consoane într-o propoziție
19. determină numărul de cifre într-o propoziție
20. determină numărul de semne de punctuație dintr-o propoziție
21. determină cel mai lung cuvânt dintr-o propoziție
22. determină cuvântul care se repetă de cele mai multe ori într-o propoziție

**Lucrarea de Laborator nr. 6**

**Tema: Pointeri (tipul indicator)**

**Scopul:** utilizarea pointeri pentru scrierea programelor în limbajul de programare C.

**Sarcini**

Scrieți un program care prelucrează pointerii așa cum este indicat în variantă, apoi afișează pe ecran rezultatul (nu uitați că memoria alocată dinamică trebuie eliberată la sfârșitul programului):

1. calculează media aritmetică a elementelor unui pointer
2. calculează de câte ori numărul „10” se întâlnește într-un pointer
3. calculează suma elementelor pare ale unui pointer
4. calculează suma elementelor impare ale unui pointer
5. calculează produsul elementelor impare ale unui pointer
6. calculează produsul elementelor pare ale unui pointer
7. determină câte elemente pare conține un pointer
8. determină câte elemente divizibile cu 3 are un pointer
9. calculează de câte ori numărul „13” se întâlnește într-un pointer
10. determină câte elemente impare conține un pointer
11. determină câte elemente nenegative conține un pointer
12. determină câte elemente pozitive conține un pointer
13. determină câte elemente divizibile cu 4 are un pointer
14. calculează suma elementelor de pe poziții pare într-un pointer
15. calculează produsul elementelor de pe poziții impare într-un pointer
16. determină câte elemente egale cu cel maximal sunt într-un pointer
17. determină câte elemente egale cu cel minimal sunt într-un pointer
18. determină câte elemente mai mici decât cel maximal sunt într-un pointer
19. determină câte elemente mai mari decât cel minimal sunt într-un pointer
20. determină câte elemente mai mici decât cel maximal sunt într-un pointer
21. determină câte elemente mai mari decât media aritmetică a tuturor elementelor sunt într-un pointer
22. determină câte elemente mai mici decât media aritmetică a tuturor elementelor sunt într-un pointer

**Lucrarea de Laborator nr. 7**

**Tema: Structuri**

**Scopul:** utilizarea structurilor și vectorilor sau pointerilor de la structuri pentru scrierea programelor în limbajul de programare C.

**Sarcini**

Scrieți un program care permite utilizatorului să selecteze unul din punctele meniului (crearea vectorului din structuri; afișarea conținutului vectorului; adăugarea datelor; modificarea datelor; eliminarea datelor; căutarea datelor conform căruiva criteriu; sortarea datelor; ieșirea din program ș.a.). Tema, pe baza căreia trebuie să creați structura, precum și câmpurile obligatorii, sunt indicate în variantă

1. Studenți (nr, nume, prenume, telefon, specialitate, an de studiu, notă)
2. Cântece (denumire, gen, interpret, durată, autor versuri, autor muzică, an)
3. Angajați (nr, nume, prenume, funcție, adresă, telefon, salariu)
4. Salon de automobile (nr, marcă, model, culoare, țară, preț)
5. Salon de telefoane mobile (IMEI, marcă, model, culoare, țară, preț)
6. Produse alimentare (nr, denumire, dată producere, termen valabilitate, preț)
7. Filme (nr, denumire, gen, regizor, durată, țară)
8. Stație auto (nr. rută, firmă, oră sosire, oră plecare, destinație, tip transport (autocar, microbuz), preț)
9. User (nr. de ordine, e-mail, nume, prenume, parolă, tip cont)
10. Elevi (nr, nume, prenume, adresă, telefon, profil, clasă, notă)
11. Magazin Internet (nr, denumire produs, unitate măsură, preț, dată primire)
12. Uzină de conserve (nr, denumire, tip, masă, preț, dată fabricare)
13. Tehnică de calcul (nr, denumire, model, preț, an editare, țară)
14. Caracter de joc (nr, nume, nick, rasă, clasă, gen, vârstă, nr. arme, nivel)
15. Bănci (denumire, adresă, director, pagină- web, curs Euro, curs Dolar)
16. Actori (nume, prenume, nr. filme, filme celebre, gen preferat, țară)
17. Sisteme de operare (denumire, subtipuri, preț, companie producătoare, procent utilizare în lume, an apariție)
18. Țări (abreviere din trei litere, populație (se poate în milioane), capitală, continent, formă de guvernare, conducător (rege, președinte, prim- ministru))
19. Genuri de sport (denumire, puncte pentru victorie, pentru remiză și pentru înfrângere, periodicitate desfășurare campionat mondial)
20. Ceasuri (firmă, model, tip (inteligent, electronic, mecanic), preț, țară, an)
21. Câini (poreclă, rasă, vârstă, culoare)
22. Televizor (firmă, model, preț, țară, culoare, termen de garanție)

**Lucrarea de Laborator nr. 8**

**Tema: Fișiere**

**Scopul:** utilizarea fișierelor pentru scrierea programelor în limbajul de programare C și C++.

**Sarcini**

Scrieți un program care permite utilizatorului să selecteze unul din punctele meniului (crearea fișierului din structuri; afișarea conținutului fișierului; adăugarea datelor în fișier; modificarea datelor din fișier; eliminarea datelor din fișier; căutarea datelor în fișier conform căruiva criteriu; sortarea datelor din fișier; ieșirea din program ș.a.). Tema, pe baza căreia trebuie să creați fișierul este aceiași ca și la lucrarea de laborator nr. 7. Se recomandă să utilizați fișiere binare.

**Lucrarea de Laborator nr. 9**

**Tema: Liste liniare**

**Scopul:** utilizarea listelor liniare pentru scrierea programelor în limbajul de programare C++.

**Sarcini**

Scrieți un program care permite utilizatorului să selecteze unul din punctele meniului (crearea listei liniare din structuri; afișarea conținutului listei; adăugarea datelor în listă; modificarea datelor din listă; eliminarea datelor din listă; căutarea datelor în listă conform căruiva criteriu; sortarea datelor din listă; ieșirea din program ș.a.). Tema, pe baza căreia trebuie să creați lista liniară este aceiași ca și la lucrarea de laborator nr. 7.

**Lucrarea de Laborator nr. 10**

**Tema: Clase**

**Scopul:** crearea claselor în limbajul de programare C++.

**Sarcini**

Scrieți un program care permite crearea unei clase simple la tema indicată mai jos, având ca funcții (metode) citirea și afișarea fiecărui câmp, precum și o funcție care calculează valoarea unui câmp (ca exemplu, de convertit prețul din EURO în lei). Clasa trebuie să conțină cel puțin trei tipuri de constructori (unul trebuie să fie fără parametri, altul- cu toți parametrii indicați, iar al treilea- cu doar câțiva parametri indicați, în dependență de temă). Tema, pe baza căreia trebuie să creați clasa este aceiași ca și la lucrarea de laborator nr. 7. Supraîncărcați cel puțin doi operatori (de exemplu „==” pentru verificarea egalității și „+” pentru adunarea a două obiecte ale clasei sau a unui obiect al clasei cu un număr).

**Lucrarea de Laborator nr. 11**

**Tema: Moștenire**

**Scopul:** utilizarea moștenirii la lucrul cu clasele în limbajul de programare С++.

**Sarcini**

Scrieți un program care permite crearea unei clase care moștenește câmpurile și metodele clasei create la lucrarea de laborator precedentă, dar mai adaugă și careva proprietăți proprii, creând și constructorii respectivi (care să corespundă constructorilor clasei de bază), apoi mai creați o clasă care moștenește caracteristicile clasei a doua, având și constructorii necesari. Tema, pe baza căreia trebuie să creați clasele derivate, precum și câmpurile noi obligatorii, sunt indicate în variantă, dar se permite de creat și alte clase derivate, care să fie logic legate cu cea de bază, în dependență de imaginația studenților.

1. Masteranzi (denumire specializare, temă a tezei de master), respectiv Doctoranzi (cod specializare, conducător al tezei de doctor)
2. Cântece pentru Show-Business (popularitate interpret), Rap (nr. descărcări)
3. Administrație (categorie de manager, nr. persoane aflate în conducere), respectiv Proprietari (tip proprietar, procent de acțiuni de care dispune)
4. Automobile care au Cutie Automată de Viteze (număr viteze, număr viteze reverse), respectiv Automobile cu GPS și cutie automată (tip GPS)
5. Telefoane Mobile cu 4G (viteză Internet, prezență wi- fi), respectiv Telefoane Mobile cu 4G cu Viteze de Internet Maxime (tehnologie utilizată)
6. Lactate (firmă producătoare), respectiv Brânzeturi (procent grăsime, tip)
7. Filme Documentare (domeniu, prezență elemente artistice), respectiv Filme (show- uri) Real- Time (tema, actori)
8. Rute Internaționale (țară destinatară, valută a țării destinatare), respectiv Rute cu Pornire și Destinație Internaționale (țară pornire, valută a țării de pornire)
9. Grupuri de Utilizatori (denumire grup, drepturi), respectiv Administrator (valabilitate parolă)
10. Elevi din Școli Profesionale (specialitate, bursă), respectiv Elevi din Colegii (cod specialitate, denumire specialitate)
11. Magazin Internet de Electrocasnice (domeniu, subdomeniu), respectiv Magazin Internet de Calculatoare (frecvență procesor, memorie operativă)
12. Uzină de Conserve Destinate Exportării peste Hotare (țară destinatară, valută a țării destinatare), respectiv Uzină de Conserve din Produse Naturale Destinate Exportării peste Hotare (denumire produs natural, termen păstrare)
13. Laptopuri (firmă producătoare, diagonală ecran), respectiv Tablete (sistem de operare, rezoluție cameră Web)
14. Caracter de Joc 1 și, respectiv, 2 (două clase la alegerea studentului)
15. Bănci de Creditare (manager sector creditare), respectiv Bănci de Acordare Credite pentru Agricultură (procent credit, termen acordare)
16. Actori de la Hollywood (onorariu mediu pentru filmare), respectiv Actori Comici de la Hollywood (regizorul preferat)
17. Microsoft Windows (versunile cele mai cunoscute), respectiv Windows NT (periodicitate apariție versiuni noi sau actualizări)
18. Țări din Europa (suprafață, valută), respectiv Țări din Uniunea Europeană (an aderare la UE)
19. Sporturi în echipă (nr. jucători echipă), respectiv Sporturi cu Mingea (tipul mingii, masa)
20. Ceasuri Inteligente (tehnologie utilizată, sisteme de operare cu care este compatibil), respectiv Smartwatch de la Samsung (perioada garanției (în luni), prezență Allways on Display)
21. Câini de rasă (preț, țară), respectiv Câini de Rasă de Elită (stăpâni precedenți, durata medie a vieții)
22. Televizor Smart (tehnologie wi-fi, prezență 3D), Televizor Smart cu 4K (rezoluție)

**Lucrarea de Laborator nr. 12**

**Tema: Clase Abstracte și Polimorfism**

**Scopul:** utilizarea claselor abstracte în limbajul de programare С++.

**Sarcini**

Scrieți un program care creează o clasă abstractă, apoi utilizează această clasă în procesul de moștenire și o realizează pe deplin. Prezentați și un exemplu de polimorfism.

1. Arie Figuri Geometrice (lungime, lățime, metoda abstractă aria()) și clasele derivate Arie Dreptunghi și Arie Pătrat.
2. Cutie (lungime, lățime, înălțime și metoda abstractă volum()) și clasele derivate Cutie în Formă de Paralelipiped Drept și Cutie în Formă de Cub.
3. Lungime Figuri Geometrice (lungime, lățime, metoda abstractă lungimea()) și clasele derivate Lungime Dreptunghi și Lungime Triunghi Isoscel.
4. Schimb Valutar (sumă, curs, metoda abstractă conversie()) și clasele derivate Schimb Lei în Euro și Schimb Euro în Lei (în primul caz suma se consideră a fi în lei, iar în al doilea caz- în Euro).
5. Convertire Lungime (lungime, metoda abstractă convertire()) și clasele derivate Convertire Centimetri în Inch și Convertire Inch în Centimetri (în primul caz lungimea se consideră a fi în cm, iar în al doilea caz- în inch).
6. Credit (sumă inițială, plată lunară și metoda abstractă sumăTotală()) și clasele derivate Credit pentru 12 Luni și Credit pentru 24 de Luni. Metoda abstractă trebuie să afișeze suma totală a produsului, știind suma achitată inițial, plata lunară și termenul de achitare.
7. Depozit (sumă inițială, rată, metoda abstractă sumăFinală()) și clasele derivate Depozit pentru 12 Luni și Depozit pentru 6 Luni. Metoda abstractă trebuie să calculeze suma finală a clientului pentru banii depuși în perioada indicată cu procentul indicat. Rata dobânzii se consideră fixă pe toată perioada.
8. Tipografie (număr pagini, preț tipărire pagină monocromă, metoda abstractă cost()) și clasele derivate Ziar Color și Ziar Monocrom. Costul tipăririi unei pagini color se consideră de patru ori mai mare decât a unei pagini alb- negru.
9. Preț Apartament (suprafață odaie, preț m2, metoda abstractă cost()) și clasele derivate Apartament cu O Odaie și Apartament cu Două Odăi. Metoda abstractă trebuie să calculeze prețul apartamentului, dacă considerăm că el are una sau două odăi (se consideră că bucătăria împreună cu coridorul și blocul sanitar au aceiași mărime ca și odăile), știind prețul unui metru pătrat și suprafața odăii.
10. Volum Corpuri Geometrice Conice (rază, înălțime, metoda abstractă volum()) și clasele derivate Volum Con și Volum Cilindru.
11. Perimetru Figuri Geometrice (lungime, lățime, metoda abstractă perimetru()) și clasele derivate Perimetru Dreptunghi și Perimetru Triunghi.