1. Explicaţi şi exemplificaţi termenul de *constructor* în Java.

|  |  |
| --- | --- |
| **Java Constructor** | **Metoda Java** |
| Un constructor este folosit pentru a inițializa starea unui obiect. | O metodă este utilizată pentru a expune comportamentul unui obiect. |
| Un constructor nu trebuie să aibă un tip de returnare. | O metodă trebuie să aibă un tip return. |
| Constructorul este invocat implicit. | Metoda este invocată în mod explicit. |
| Compilatorul Java oferă un constructor implicit dacă nu aveți niciun constructor într-o clasă. | Metoda nu este furnizată de compilator în niciun caz. |
| Numele constructorului trebuie să fie același cu numele clasei. | Numele metodei poate fi sau nu același cu numele clasei. |

Exemplu de constructor: Fish myFish = **new** Fish();

1. Explicaţi termenii *atribut/metodă publică* şi *atribut/metodă privată* în legătură cu noțiunea de clasă.  
   public si private sunt nivede de acces ai atributului/metodei,  
   public inseamna ca e disponibil accesul catre acest atribus/metoda, daca cuvintul public lipseste nu se poate determina tipul acelui atricut/metoda  
   private se utilizeaza pentru a ascunde atributele/metodele de restul programului, si sunt chemate doar in corpul clasei create.
2. Cum se propagă excepţiile? Daţi un exemplu scurt.  
   try {

int X,Y,Z;

//cod unde se face ceva cu valori numerici

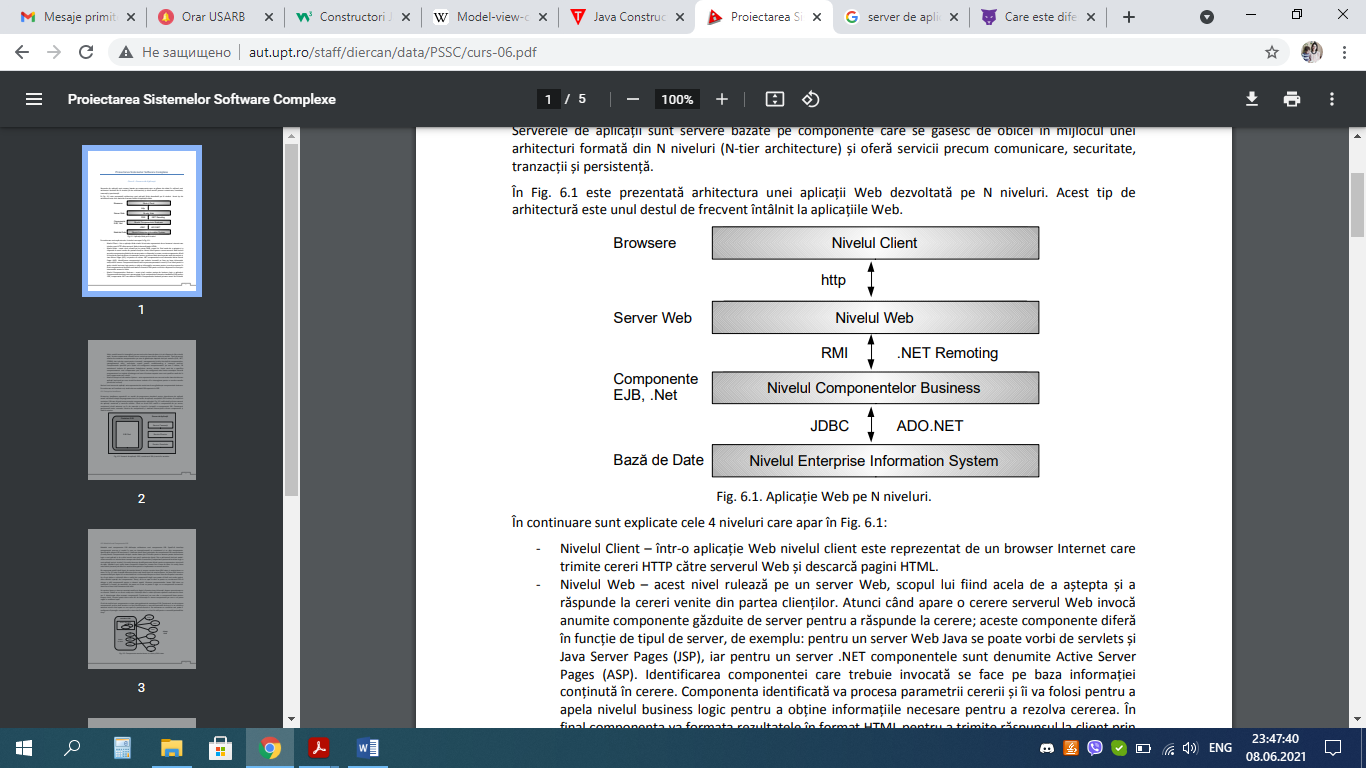
}catch(InputMismatchException e) {

System.out.println("Introduceti va rog o valoare numerica");

}

1. Care este rolul controller-ului în arhitectura MVC?

**Controler** Cu acest element putem controla accesul la aplicația noastră. Pot fi fișiere, scripturi (eng. scripts) sau programe, in general orice tip de informație permisă de interfață. În acest fel putem diversifica conținutul nostru de o formă dinamică și statică, în același timp.

1. Ce este testarea unitară (a unităților)?  
   Testarea unitara JUnit are ca scop modalitatea de a fixa functionalul si daca se fac modificari in cod care strica functoinalul acestuia, atunci testele ne dau de stire ca ceva nu-i bine. Cu ajutorul testelor fixam anumite situatii.
2. Care sunt principalele diferenţe între o aplicaţie de sine stătătoare şi o aplicație web?  
   O aplicație de consolă rulează la consola de prompt de comandă. Nu are interfață grafică doar text simplu. De obicei, acestea sunt pentru utilități care nu necesită multă interacțiune.  
   O aplicație web este o aplicație care rulează pe un server web și servește HTML și este accesată printr-un browser web.
3. Definiți serverul de aplicații. Descrieți schematic arhitectura unei aplicații enterprise. Prin ce se deosebește un server de aplicații față de un server web? Ce servere de aplicații cunoașteți?  
   Serverele de aplicații sunt servere bazate pe componente care se găsesc de obicei în mijlocul unei arhitecturi formată din N niveluri și oferă servicii precum comunicare, securitate, tranzacții și persistență.  
     
   Un server Web poate fi fie un program de calculator, fie un computer care rulează un program care este responsabil de acceptarea cererilor HTTP de la clienți, care servește răspunsuri HTTP înapoi împreună cu conținut de date opționale, care de obicei sunt pagini web, cum ar fi documente HTML și obiecte legate pe acesta. Un server de aplicații este genul de motor software care va livra diverse aplicații unui alt dispozitiv. Este genul de computer găsit într-o rețea de birou sau universitate care permite tuturor celor din rețea să ruleze software de pe același aparat. Un server Web poate fi fie un program de calculator, fie un computer care rulează un program care este responsabil de acceptarea cererilor HTTP de la clienți, care servește răspunsuri HTTP înapoi împreună cu conținut de date opționale, care de obicei sunt pagini web, cum ar fi documente HTML și obiecte legate pe acesta. Un server de aplicații este genul de motor software care va livra diverse aplicații unui alt dispozitiv. Este genul de computer găsit într-o rețea de birou sau universitate care permite tuturor celor din rețea să ruleze software de pe același aparat.  
   **Exemple**: Server de aplicații Java Java, server weblogic, Apache Geronimo, IBM WebSphere Application Server, Glass Fish Server.
4. Definiți Maven. Enumerați avantajele utilizării Maven. Descrieți cum sunt gestionate dependințele într-un proiect Maven vs. un proiect care nu utilizează Maven. Cunoașteți alternative ale Maven?  
   Maven este un sistem de build și administrare a proiectelor, descriere procesul de build al software-ului și descriere dependențele acestuia.  
   POM-ul principal conține informații despre module, precum și despre dependențele proiectului. Maven este capabil să identifice dependențele tranzitive și să includă toate bibliotecile care depind de cea cerută.

1. Elaborați o aplicație web pentru conversia din mile în km (și invers). Interacțiunea cu utilizatorul va fi realizată printr-un formular web. Aplicația trebuie să folosească modelul MVC. Datele din formular trebuie să fie supuse validării și mesajele corespunzătoare de eroare trebuie să fie afișate în caz de date eronate. Toate clasele din aplicație trebuie să fie acoperite cu teste unitare.