জাভাতে **এনক্যাপসুলেশন (Encapsulation)** হলো অবজেক্ট-ওরিয়েন্টেড প্রোগ্রামিং (OOP) এর একটি মৌলিক কনসেপ্ট, যা ডেটা এবং মেথডগুলিকে একটি একক ইউনিট বা ক্লাসের মধ্যে আবদ্ধ (encapsulate) করে এবং ডেটার নিরাপত্তা নিশ্চিত করে। এনক্যাপসুলেশনের মূল উদ্দেশ্য হলো ডেটাকে বাইরের অবাঞ্ছিত অ্যাক্সেস থেকে রক্ষা করা এবং ডেটা ম্যানিপুলেশন কন্ট্রোল করা।

**এনক্যাপসুলেশনের মূল ধারণা:**

1. **ডেটা হাইডিং (Data Hiding)**: ক্লাসের ভেরিয়েবল বা ডেটাকে প্রাইভেট (private) ঘোষণা করে বাইরের ক্লাস বা অবজেক্ট থেকে সরাসরি অ্যাক্সেস বন্ধ করে দেওয়া।
2. **পাবলিক ইন্টারফেস**: ডেটা অ্যাক্সেস বা মডিফাই করার জন্য পাবলিক মেথড (getter এবং setter) প্রদান করা।
3. **ডেটা সুরক্ষা**: ডেটা যাতে অবৈধ বা ভুলভাবে পরিবর্তন না হয়, তা নিশ্চিত করা।

**এনক্যাপসুলেশন কিভাবে কাজ করে?**

* **প্রাইভেট ভেরিয়েবল**: ক্লাসের ভেরিয়েবলগুলোকে private ঘোষণা করা হয়, যাতে সেগুলো সরাসরি বাইরে থেকে অ্যাক্সেস করা না যায়।
* **পাবলিক মেথড**: এই প্রাইভেট ভেরিয়েবলগুলোর মান পড়া (getter) বা পরিবর্তন করা (setter) এর জন্য পাবলিক মেথড ব্যবহার করা হয়।
* **কন্ট্রোলড অ্যাক্সেস**: setter মেথডের মাধ্যমে ডেটা ভ্যালিডেশন বা শর্ত যুক্ত করে ডেটা সুরক্ষিত রাখা যায়।

**উদাহরণ:**

java

Copy

class Student {

// প্রাইভেট ভেরিয়েবল (ডেটা হাইডিং)

private String name;

private int age;

// পাবলিক getter মেথড (ডেটা পড়ার জন্য)

public String getName() {

return name;

}

// পাবলিক setter মেথড (ডেটা সেট করার জন্য)

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

if (age > 0) { // ডেটা ভ্যালিডেশন

this.age = age;

} else {

System.out.println("Invalid age!");

}

}

}

public class Main {

public static void main(String[] args) {

Student student = new Student();

// setter মেথড ব্যবহার করে ডেটা সেট করা

student.setName("John");

student.setAge(20);

// getter মেথড ব্যবহার করে ডেটা পড়া

System.out.println("Name: " + student.getName());

System.out.println("Age: " + student.getAge());

// অবৈধ ডেটা সেট করার চেষ্টা

student.setAge(-5); // এটি "Invalid age!" প্রিন্ট করবে

}

}

**উপরের কোডের ব্যাখ্যা:**

1. **প্রাইভেট ভেরিয়েবল**: name এবং age ভেরিয়েবলগুলো private ঘোষণা করা হয়েছে, তাই এগুলো সরাসরি Student ক্লাসের বাইরে থেকে অ্যাক্সেস করা যায় না।
2. **getter এবং setter মেথড**: getName(), setName(), getAge(), এবং setAge() মেথডগুলো পাবলিক, তাই এগুলো ব্যবহার করে ডেটা অ্যাক্সেস বা মডিফাই করা যায়।
3. **ডেটা ভ্যালিডেশন**: setAge() মেথডে একটি শর্ত যুক্ত করা হয়েছে, যাতে বয়স নেগেটিভ হতে না পারে।

**এনক্যাপসুলেশনের সুবিধা:**

1. **ডেটা সুরক্ষা**: ডেটা সরাসরি অ্যাক্সেস করা যায় না, তাই ডেটা সুরক্ষিত থাকে।
2. **ফ্লেক্সিবিলিটি**: ডেটা ম্যানিপুলেশন কন্ট্রোল করা যায় এবং ভবিষ্যতে পরিবর্তন করা সহজ হয়।
3. **রিইউজেবিলিটি**: এনক্যাপসুলেটেড কোড অন্য প্রোগ্রামে পুনরায় ব্যবহার করা যায়।
4. **ডিবাগিং সহজ**: ডেটা অ্যাক্সেস বা পরিবর্তন শুধুমাত্র নির্দিষ্ট মেথডের মাধ্যমে হয়, তাই ডিবাগ করা সহজ।

**সারমর্ম:**

এনক্যাপসুলেশন হলো জাভার একটি শক্তিশালী ফিচার, যা ডেটা এবং মেথডগুলিকে একটি ক্লাসের মধ্যে আবদ্ধ করে এবং ডেটা সুরক্ষা নিশ্চিত করে। এটি কোডকে আরও সুরক্ষিত, ফ্লেক্সিবল এবং রিইউজেবল করে তোলে।