**Exercise 3: Implementing the Builder Pattern**

**Computer.java**

public class Computer {

private final String cpu;

private final String ram;

private final String storage;

private final String gpu;

private final String os;

private final boolean wifi;

private final boolean bluetooth;

private Computer(Builder b) {

cpu = b.cpu;

ram = b.ram;

storage = b.storage;

gpu = b.gpu;

os = b.os;

wifi = b.wifi;

bluetooth = b.bluetooth;

}

public String toString() {

return """

--- Computer ---

CPU: %s

RAM: %s

Storage: %s

GPU: %s

OS: %s

WiFi: %s

Bluetooth: %s

""".formatted(cpu, ram, storage, gpu, os, wifi, bluetooth);

}

public static class Builder {

private String cpu;

private String ram;

private String storage;

private String gpu;

private String os = "FreeDOS";

private boolean wifi;

private boolean bluetooth;

public Builder cpu(String v) { cpu = v; return this; }

public Builder ram(String v) { ram = v; return this; }

public Builder storage(String v) { storage = v; return this; }

public Builder gpu(String v) { gpu = v; return this; }

public Builder os(String v) { os = v; return this; }

public Builder wifi() { wifi = true; return this; }

public Builder bluetooth() { bluetooth = true; return this; }

public Computer build() {

if (cpu == null || ram == null || storage == null)

throw new IllegalStateException("cpu, ram, storage required");

return new Computer(this);

}

}

}

**ComputerDemo.java**

public class ComputerDemo {

public static void main(String[] args) {

Computer gamingRig = new Computer.Builder()

.cpu("AMD Ryzen 9")

.ram("64GB")

.storage("2TB SSD")

.gpu("RTX 4090")

.os("Windows 11 Pro")

.wifi()

.bluetooth()

.build();

Computer officePC = new Computer.Builder()

.cpu("Intel Core i5")

.ram("16GB")

.storage("512GB SSD")

.gpu("Integrated")

.os("Ubuntu")

.wifi()

.build();

Computer budgetPC = new Computer.Builder()

.cpu("Intel Pentium")

.ram("4GB")

.storage("128GB HDD")

.gpu("None")

.build();

System.out.println(gamingRig);

System.out.println(officePC);

System.out.println(budgetPC);

}

}

**Output:**

**A computer screen shot of a black screen

AI-generated content may be incorrect.**