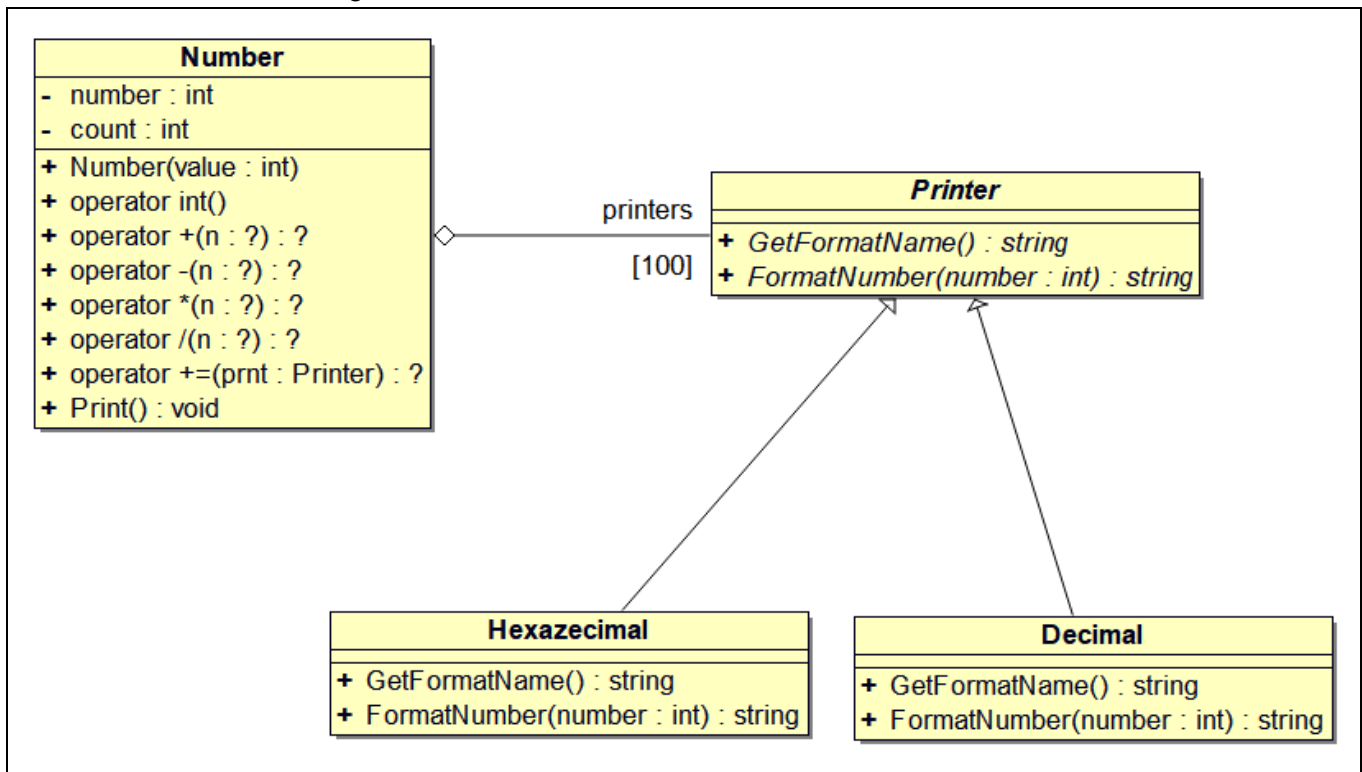


2. Fie urmatoarea diagrama UML:



Sa se implementeze diagrama de clase astfel încât codul din main() descris mai jos:

COD main()

```
int main()
{
    Number n1(1), n2(2), n3(10);
    Number n4( (int)((((n1 + n2)*n3) / n2) + (n1 - n2)) );
    (n4.operator+=(new Hexazecimal())) += (new Decimal());
    n4.Print();
    std::cout << "Valoarea hexazecimala este: " << /* completati pentru a afișa în baza 16*/ << (int)n4 << std::endl;
    return 0;
}
```

să afișeze:

```
Format: Hexazecimal Value: 0x0000000E
Format: Decimal Value: 14
Valoarea hexazecimala este: e
```

Barem corectare (30 pct)

Cerinta	Punctaj
Organizarea programului în 8 fișiere ("main.cpp", "number.cpp", "number.h", "Hexazecimal.cpp", "Hexazecimal.h", "Decimal.cpp", "Decimal.h", "Printer.h")	3 pct
Se respecta exact datele din diagrama UML (nu apar metode sau date membru adiționale la clasele de mai sus).	2 pct
Operatorul de castare la int	1 pct
Operatorii +, -, * si / din clasa <i>Number</i> (1 pct per operator)	4 pct
Metoda <i>GetFormatName</i> din clasele Hexazecimal si Decimal (1.5 pct per clasa)	3 pct
Metoda <i>FormatNumber</i> din clasele Hexazecimal si Decimal (3pct per clasa)	6 pct
Definirea corecta a clasei <i>Printer</i>	1 pct
Operatorul += din clasa <i>Number</i>	4 pct
Metoda <i>Print</i> din clasa <i>Number</i>	4 pct
Constructorul clasei <i>Number</i>	1 pct
Completarea codului din main din std::cout pentru a afișa valoarea din n4 în forma hexazecimala	1 pct

OBS:

- Aveți voie sa folosit funcțiile din librăriile standard
- **NU** aveți voie sa adaugati metode sau date membru în plus fata de cele explicate în diagrame UML. Adaugarea acestora duce la nepunctarea cerintei unde se aplica modificarea adaugata.
- **NU** aveți voie sa utilizati tool-uri de conversie automata din UML in C++
- Unde observati în diagrama UML caracterul ? (semnul întrebării) se referă la un tip de date pe care trebuie sa il deduceti voi (pentru un parametru sau tip de return)