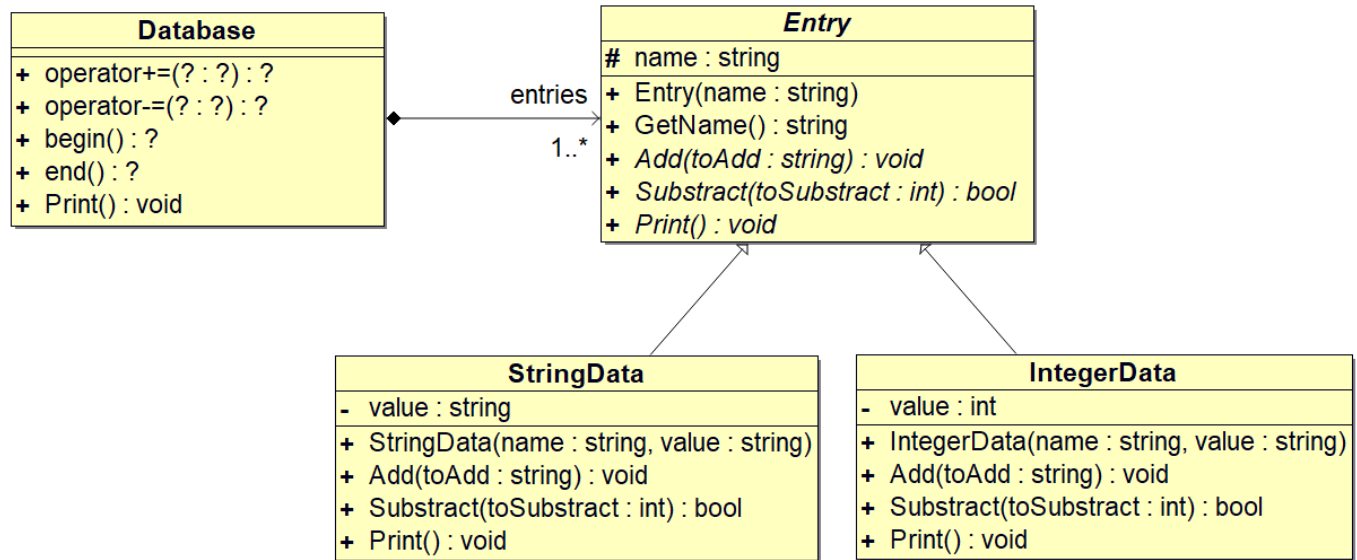


3. Fie următoarea diagramă UML:



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

#### COD main()

```

int main() {
    Database d;
    Entry* e1 = (Entry*)new StringData("Student", "Alexandru");
    Entry* e2 = (Entry*)new IntegerData("NotaEngleza", "10");
    (d += e1) += (new StringData("Facultate", "FII"));
    (d += e2) += (new IntegerData("NotaP00", "9"));
    d.Print();
    for (Entry* e : d) {
        e->Subtract(5);
    }
    d.Print();
    d -= "NotaEngleza";
    d.Print();
    for (auto e : d) {
        e->Add("1");
    }
    d.Print();
    return 0;
}
  
```

să afișeze:

```

Student = Alexandru;Facultate = FII;NotaEngleza = 10;NotaP00 = 9;
StringData: len(FII) < 5
Student = Alex;Facultate = FII;NotaEngleza = 5;NotaP00 = 4;
Student = Alex;Facultate = FII;NotaP00 = 4;
Student = Alex1;Facultate = FII1;NotaP00 = 5;
  
```

**Barem corectare (30 pct)**

Cerinta	Punctaj
Organizarea programului în 9 fișiere ("main.cpp", "Entry.cpp", "Entry.h", "StringData.cpp", "StringData.h", "IntegerData.cpp", "IntegerData.h", "Database.h", "Database.cpp")	3 pct
Se respecta exact datele din diagrama UML (nu apar metode sau date membru adiționale la clasele de mai sus).	2 pct
Metodele Add, Subtract și Print din clasa Entry (1 pct fiecare)	3 pct
Metoda GetName din clasa Entry	1 pct
Constructorul clasei IntegerData	2 pct
Metodele Subtract și Add din clasa IntegerData (2 pct fiecare)	4 pct
Metodele Subtract și Add din clasa StringData (2 pct fiecare)	4 pct
Metodele Print din clasele IntegerData și String Data (1 pct fiecare)	2 pct
Operatorul+= din clasa Database	2 pct
Operatorul-= din clasa Database	4 pct
Metodele begin și end din clasa Database (1 pct fiecare)	2 pct
Metoda Print din clasa Database	1 pct

**OBS:**

- Aveți voie sa folosiți 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 ne punctarea cerintei unde se aplica modificarea adaugata.
- **NU** aveți voie sa utilizati tool-uri de conversie automata din UML in C++
- Metoda **Add** are comportament diferit în funcție de clasa în care e implementată: pentru StringData append-ează string-ul parametru la valoarea curentă, iar pentru IntegerData string-ul este convertit la număr și este adunat la valoarea curentă (a se observa afișarea)
- Metoda **Subtract** are comportament diferit în funcție de clasa în care e implementată: pentru StringData șterge ultimele x caractere din valoare (dacă  $x > \text{len}(\text{value})$  nu se face operația), iar pentru IntegerData numărul este scăzut din valoarea curentă (a se observa afișarea)