

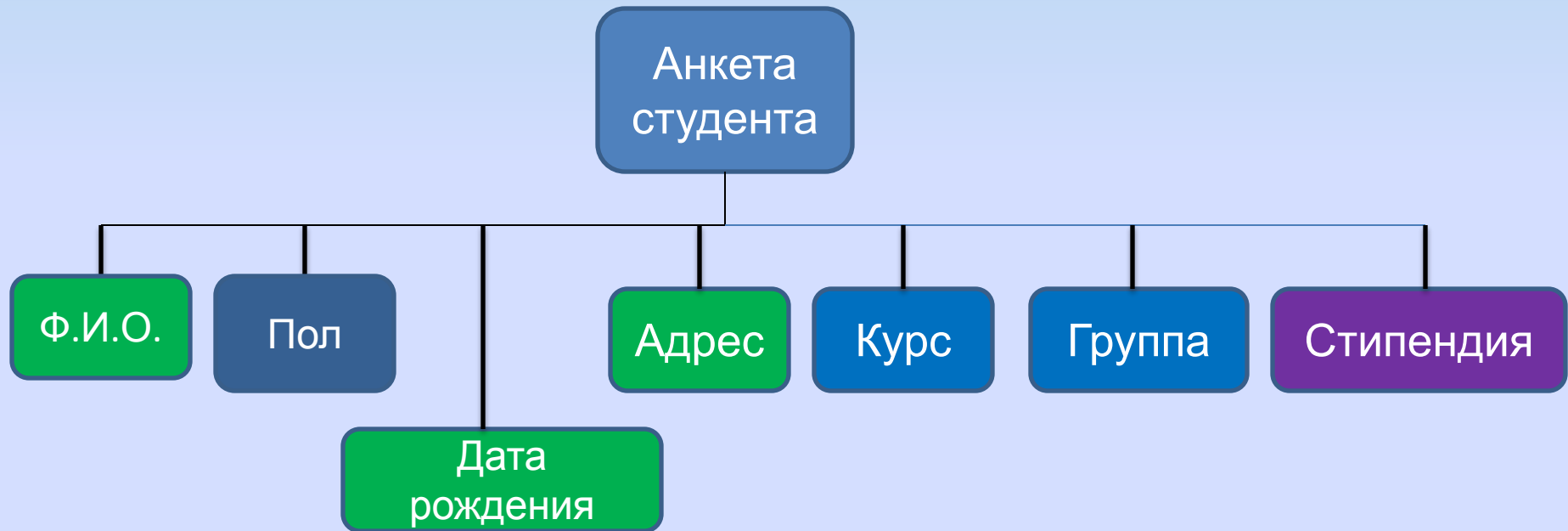
Комбинированный тип данных

ЗАПИС

И

Комбинированный тип данных - это структурированный тип, состоящий из фиксированного числа компонент (полей) разного типа. Комбинированный тип имеет еще и другое название – **запись**.

Обычно запись содержит совокупность разнотипных атрибутов, относящихся к одному объекту. Например, анкетные сведения о студенте вуза могут представлены в виде информационной структуры.



Такая структура называется **двухуровневым деревом**. В Паскале эта информация может храниться в одной переменной типа **Record (Запись)**. Задать тип и описать соответствующую переменную можно следующим образом:

```
Type Anketa1= Record  
    Fio : String[50];  
    Pol : Char;  
    Dat : String[16];  
    Adres : String[50];  
    Curs : 1 .. 5;  
    Grup 1 .. 10;  
    Stip : Real;  
    End;  
Var Student : Anketa1;
```

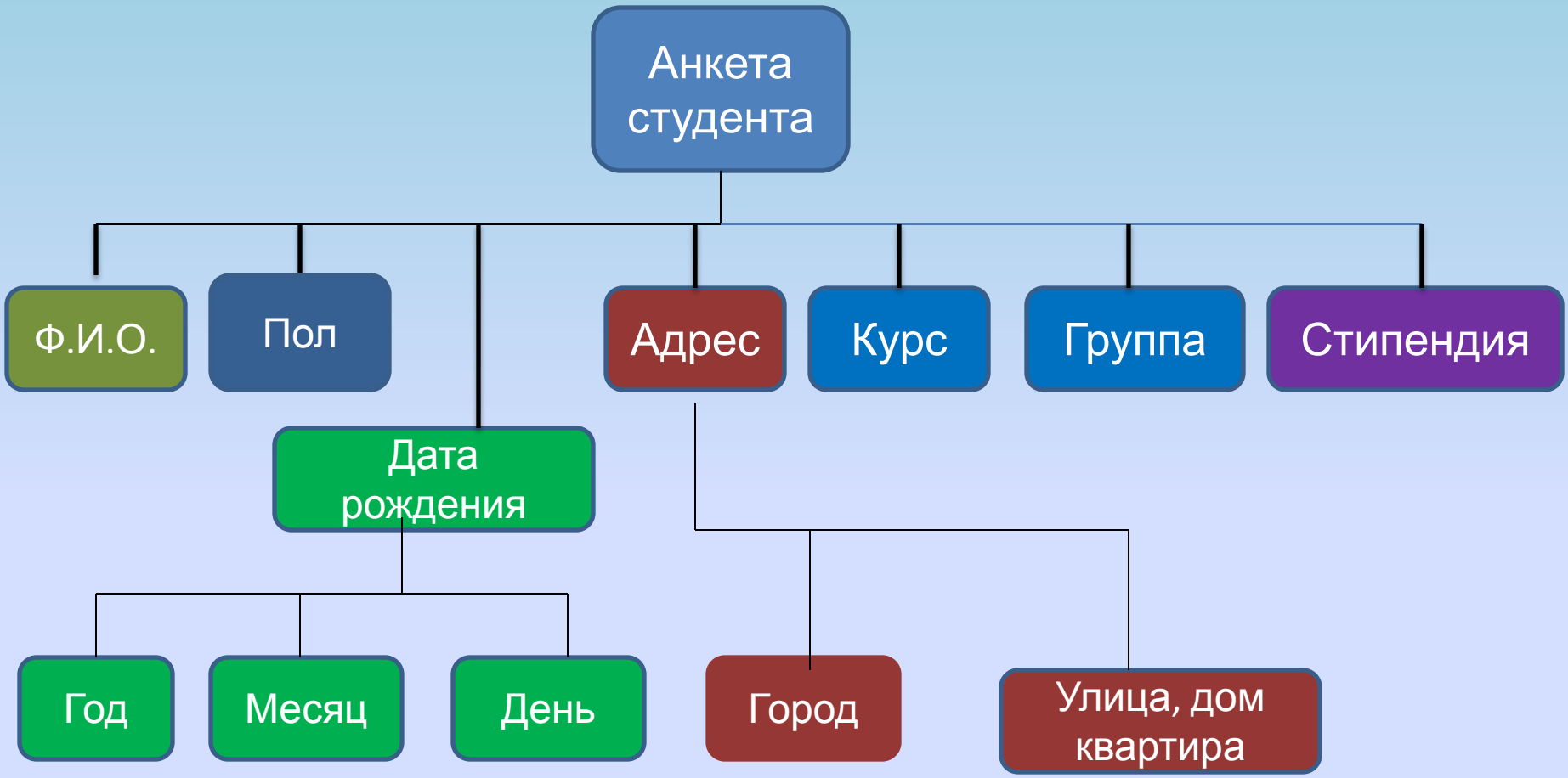
К каждому элементу записи можно обратиться, используя **составное имя**,

которое имеет следующую структуру:

< имя переменной > . < имя поля >

Например, Student . Fio; Student . Dat и т.д.

Поля записи могут иметь любой тип. В частности сами могут быть записями.



Описание соответствующей записи будет
выглядеть так:

```
Type Anket2= Record  
  Fio : String[50];  
  Pol : Char;  
  Dat : Record  
    God : Integer;  
    Mes : String[10];  
    Den : 1 .. 31  
    End;  
  Adres : Record  
    Gorod : String[20]  
    UIDomKv : String[30];  
    End;  
  Curs : 1 .. 5;  
  Grup 1 .. 10;  
  Stip : Real;  
  End;  
Var Student : Anketa2;
```

В программе могут использоваться **массивы записей**. Если на факультете 500 студентов , то все анкетные данные о них можно представить в массиве:

```
Var Student : Array [1 .. 500] of Anketa1;
```

В таком случае , например, год рождения 5-го в списке студента

хранится в переменной ***Student[5].Dat.God.***

Любая обработка записей, в том числе ввод и вывод, производится поэлементно. Например, ввод сведений о 500 студентах можно организовать так:

```
For i := 1 to 500 do  
With Student[ i ] do  
Begin  
    Write ( ‘ Ф.И.О. : ’ ) ; Readln (FIO) ;  
    Write ( ‘ Пол (м/ж) : ’ ) ; Readln (Pol) ;  
    Write ( ‘ Дата рождения : ’ ) ; Readln (Dat) ;  
    Write ( ‘ Адрес : ’ ) ; Readln (Adres) ;  
    Write ( ‘ Курс : ’ ) ; Readln (Curs) ;  
    Write ( ‘ Группа : ’ ) ; Readln (Grup) ;  
    Write ( ‘ Стипендия (руб.) : ’ ) ; Readln (Stip)  
End;
```

В этом примере использован **оператор присоединения**, который имеет следующий вид:

```
With <переменная типа запись> do
```

<оператор>;

Он позволяет , один раз указав имя переменной типа запись после слова ***With*** , работать в пределах оператора с именами полей как с обычными переменными, т.е. не писать составных имен.

Работа с файлами записей

Пример.

Сформировать файл *FM.DAT* , содержащий экзаменационную ведомость студенческой группы. Записи файла состоят из следующих элементов: *фамилия, имя, отчество, номер зачетной книжки, оценка.*

Program Examen;

Type Stud = Record

FIO : String[30];

Nz : String[6];

Mark : 2 .. 5

End;

Var Fstud: **File of** Stud; { Фай л записей }

S: Stud; { Переменная типа «Запись»}

N, i: Byte;

Begin

Assing(Fstud, 'FM.DAT'); **Rewrite**(Fstud);

Write (' Количество студентов в группе? '); **Readln** (N);

For i := 1 to N **do**

Begin

Write (i :1, '-й , Фамилия И. О. '); **Readln** (S.FIO);

Write (' Номер зачетки '); **Readln**(S.Nz);

Write (' Оценка '); **Readln**(S.Mark);

Write (Fstud, S);

End;

Writeln (' Формирование фай ла закончено ');

Close (Fstud)

End.

Пример

Имеется описание:

```
Adres : Record
```

```
    Gorod : String[20];
```

```
    Ulica:String;
```

```
    Dom,Kv : 1 .. 100;
```

```
End;
```

Используя оператор присваивания,
присвоить переменной ADR значение
«**Москва, улица Арбат, д 1, кв. 5.**»

Упражнения

- Описать запись, содержащую сведения о рейсе самолета.
- Описать запись расписания уроков на неделю.

Домашнее задание

Вычислить средний балл учеников класса, если известны оценки каждого ученика по математике, русскому языку и физике.
Распечатать список учеников, имеющих средний балл выше среднего в классе.