CMPE 152: Compiler Design

November 14 Lab

Department of Computer Engineering
San Jose State University

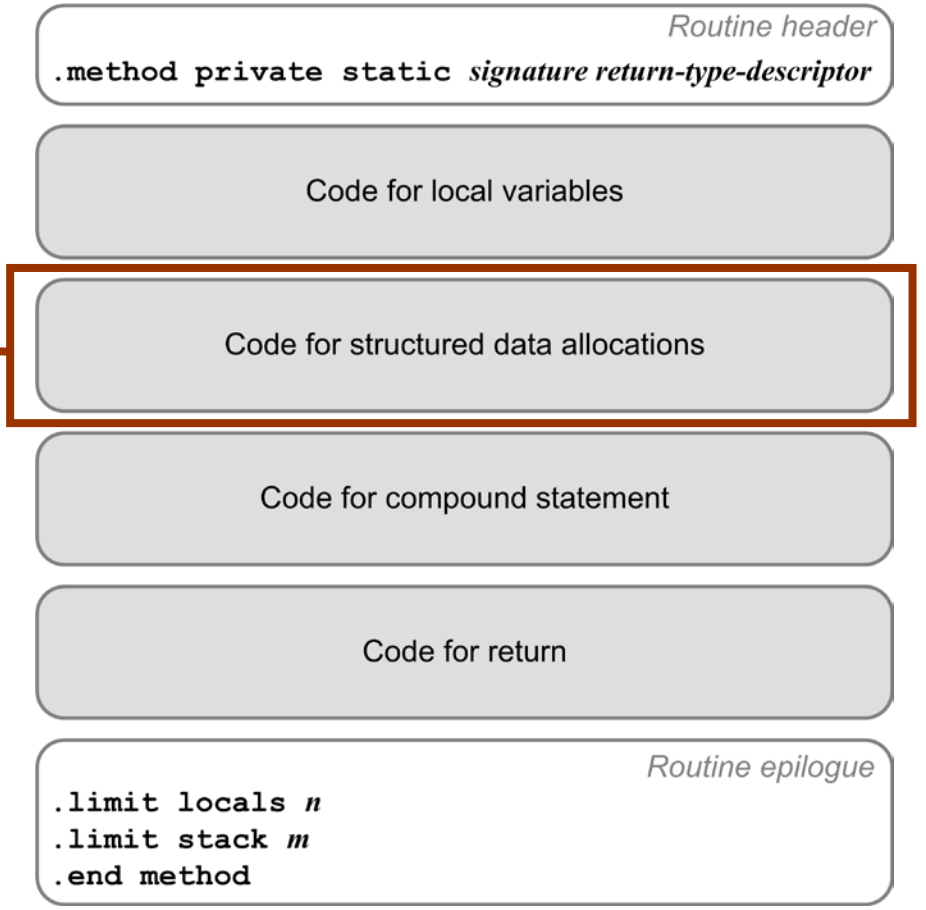
Fall 2017
Instructor: Ron Mak

www.cs.sjsu.edu/~mak

Records and Fields

- Recall the code template for a Jasmin method.

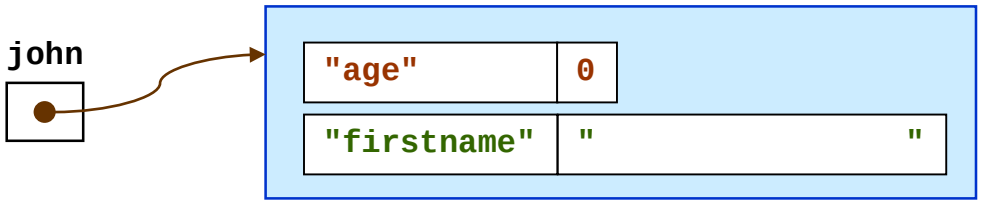
Code to allocate **records** here!



- Implement the value of <u>each</u> Pascal record variable as a **`java.util.HashMap`** object.
  - **Keys:** Field names (as strings)
  - **Values:** Field values (as objects)

Pascal Records in the JVM

- Each record value is a separate hash table.
  - **Keys:** field names
  - **Values:** field values
- Allocate and initialize each value.



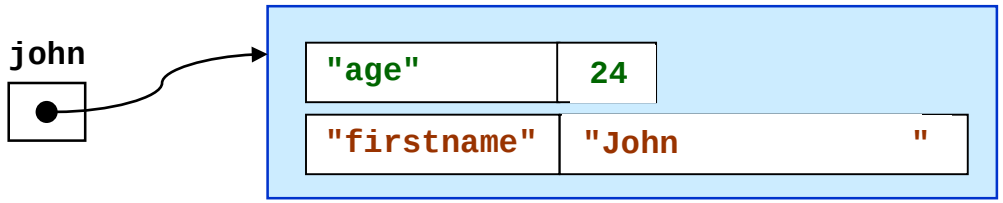

```
PROGRAM RecordTest2;

TYPE
  String16 =
    ARRAY [1..16] OF char;

  PersonRec =
    RECORD
      firstName : String16;
      age       : integer;
    END;

VAR
  john : PersonRec;

BEGIN
  ...
END.
```

```
new  java/util/HashMap
dup
invokenonvirtual  java/util/HashMap/<init>()V
dup
ldc    "age"
iconst_0
invokestatic   java/lang/Integer.valueOf(I)Ljava/lang/Integer;
invokevirtual  java/util/HashMap.put(Ljava/lang/Object;
                        Ljava/lang/Object;)Ljava/lang/Object;
pop
dup
ldc     "firstname"
bipush  16
invokestatic   PaddedString.create(I)Ljava/lang/StringBuilder;
invokevirtual  java/util/HashMap.put(Ljava/lang/Object;
                        Ljava/lang/Object;)Ljava/lang/Object;
pop
putstatic  recordtest2/john Ljava/util/HashMap;
```

Convert the **int** value 0 to an **Integer** object.

Why **pop**?

Set the Values of Record Fields

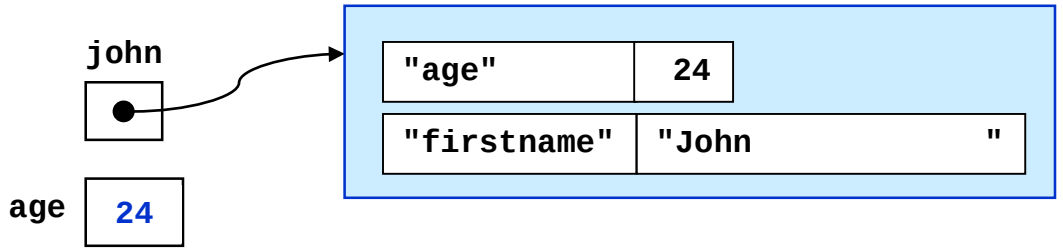



```
PROGRAM RecordTest2;

TYPE
  String16 = ARRAY [1..16]
               OF char;

  PersonRec = RECORD
                firstName
                  : String16;
                age
                  : integer;
              END;

VAR
  john : PersonRec;
  age  : integer;

BEGIN
  john.age := 24;
  john.firstName := 'John';
  age := john.age;
END.
```

```
getstatic       recordtest2/john Ljava/util/HashMap;
ldc             "age"
bipush          24
invokestatic    java/lang/Integer.valueOf(I)Ljava/lang/Integer;
invokevirtual   java/util/HashMap.put(Ljava/lang/Object;
                                      Ljava/lang/Object;)Ljava/lang/Object;
pop

getstatic       recordtest2/john Ljava/util/HashMap;
ldc             "firstname"
invokevirtual   java/util/HashMap.get(Ljava/lang/Object;)Ljava/lang/Object;
checkcast       java/lang/StringBuilder
dup
iconst_0
invokevirtual   java/lang/StringBuilder.setLength(I)V
ldc             "John"
invokevirtual   java/lang/StringBuilder.append(
                                 Ljava/lang/String;)Ljava/lang/StringBuilder;
bipush          16
iconst_4
invokestatic    PaddedString.blanks(II)Ljava/lang/StringBuilder;
invokevirtual   java/lang/StringBuilder.append(
                          Ljava/lang/CharSequence;)Ljava/lang/StringBuilder;
pop
```

Access Values of Record Fields

```
PROGRAM RecordTest2;

TYPE
  String16 = ARRAY [1..16]
               OF char;

  PersonRec = RECORD
                firstName
                  : String16;
                age
                  : integer;
              END;

VAR
  john : PersonRec;
  age  : integer;

BEGIN
  john.age := 24;
  john.firstName := 'John';
  age := john.age;
END.
```




```
getstatic      recordtest2/john Ljava/util/HashMap;
ldc            "age"
invokevirtual  java/util/HashMap.get(Ljava/lang/Object;)
                                    Ljava/lang/Object;
checkcast      java/lang/Integer
invokevirtual  java/lang/Integer.intValue()I
putstatic      recordtest2/age I
```