

EECS3311 Software Design (Fall 2020)

Q&A - Lab2

Thursday, October 1

② db.override("mak", 4b)

↳ version of override called?

↳ execute db.insert but which version?

③ create {TREE_DB} db make

④ db.inset(...)

⑤ db.override(...)

↳ db.inset

```

deferred class DATABASE[K -> COMPARABLE, V]
  feature -- Abstraction Function
  model: REL[K, V]
  deferred
  ensure
    ... -- other postcondition omitted
    all_model_pairs_exist_as_key_value_tuples: ...
  end
  feature Deferred Routines
  insert(p_key: K; p_value: V)
  require
    no_previous_entry: ...
  deferred
  ensure
    entry_added: ...
  end
  ... -- other deferred routines omitted
  feature -- Basic, intermediate, advanced
  override(p_key: K; p_value: V)
  do
    Current.insert(p, k) for illustration only
  end
  ensure
    overridden_result: ...
  end
  ... -- other effective routines omitted
end
  
```

basic building blocks

Basic, intermediate, advanced

Current.insert(p, k) for illustration only

db: DATABASE[STR, INT]

create db.make X

create {DATABASE} db.make X

create {LINEAR_DB[-, -]} db.make

① db.insert("alan", 23)

↳ postcondition where?

```

class TREE_DB[K -> COMPARABLE, V]
  inherit DATABASE[K, V]
  ...
  feature {ES_TEST} -- Restricted Attributes
  bst: BALANCED_BST[K, V] -- imp
  feature -- Abstraction Function
  model: REL[K, V]
  do
    ...
  end
  postcondition inherited from {DATABASE}.model
  end
  feature -- Implemented Routines from DATABASE
  insert(p_key: K; p_value: V)
  -- precondition inherited from {DATABASE}.insert
  do
    ...
  end
  -- postcondition inherited from {DATABASE}.insert
  end
  -- basic, intermediate, advanced features from DATABASE
  -- are all inherited verbatim without being redefined.
end
  
```

```

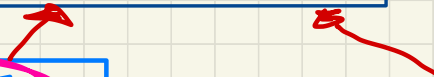
class LINEAR_DB[K -> {COMPARABLE, HASHABLE}, V]
  inherit DATABASE[K, V]
  ...
  feature {ES_TEST} -- Restricted Attributes
  keys: ARRAY[K] -- imp
  values: HASH_TABLE[V, K] -- imp
  feature -- Abstraction Function
  model: REL[K, V]
  do
    ...
  end
  postcondition inherited from {DATABASE}.model
  end
  feature -- Implemented Routines from DATABASE
  insert(p_key: K; p_value: V)
  -- precondition inherited from {DATABASE}.insert
  do
    ...
  end
  -- postcondition inherited from {DATABASE}.insert
  end
  -- basic, intermediate, advanced features from DATABASE
  -- are all inherited verbatim without being redefined.
end
  
```

```
class PARENT
  feature
    [f1]
  deferred
  end
  [f2]
  do
  end
  [f1]
end
```

```
obj : PARENT
  ...
  [obj].f2
  dynamic
  type?
```

```
class CHILD_1
  inherit PARENT
  [f1]
  do
  end
```


```
class CHILD_2
  inherit PARENT
  [f1]
  do
  end
```



```
abstract class Database {  
    :  
}
```

```
Database db =  
new Database() ;
```

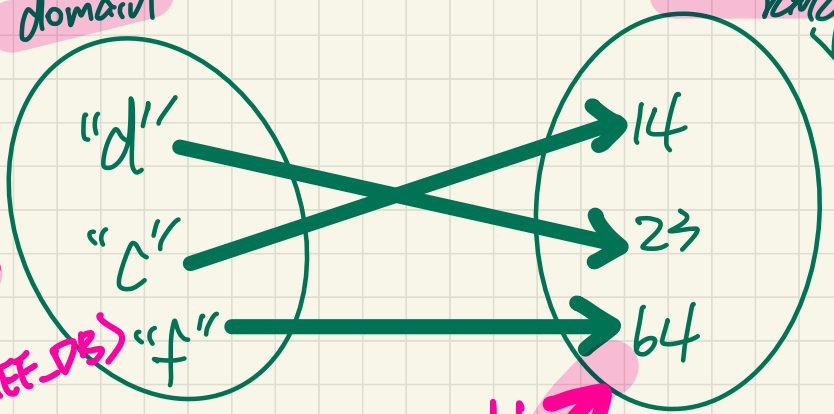
```
class TreeDB {  
    :  
}
```



① Contracts
↳ REL

domain

range



{ ("d" → 23),
("c" → 14),
("f" → 64) }

↑ specification
(model)
REL

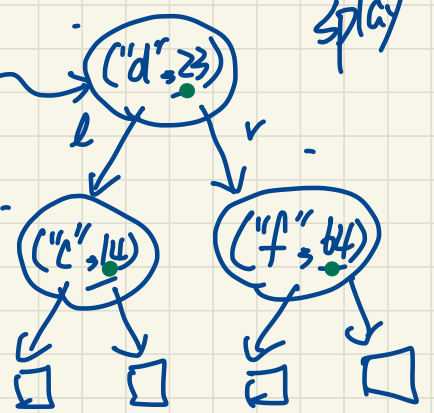
↓ implementation

model
(TREE_DB)

TREE_DB

splay tree

bst



model
(L_DB)

LINEAR_DB

keys

	1	2	3
keys	"d"	"f"	"c"

values

key	value
"c"	14
"d"	23
"f"	64