ASSIGNMENT:5TH

NAME: MUHAMMAD ISHAQ

ROLL NO: 7351

SECTION: BSSE 6TH B

SUBITTED TO: MAM MANEEHA

QUESTION 1: What is The Vienna Development Method (VDM), and why VDM was introduced.

ANSWER:

VDM (Vienna Development Method):

The Vienna Development Method (VDM) is one of the longest established model-oriented formal methods for the development of computer-based systems and software. It consists of a group of mathematically well-founded languages and tools for expressing and analyzing system models during early design stages, before expensive implementation commitments are made.

Why VDM was introduced:

The construction and analysis of the model help to identify areas of incompleteness or ambiguity in informal system specifications, and provide some level of confidence that a valid implementation will have key properties, especially those of safety or security. VDM has a strong record of industrial application, in many cases by practitioners who are not specialists in the underlying formalism or logic. Experience with the method suggests that the effort expended on formal modeling and analysis can be recovered in reduced rework costs arising from design errors.

System Modelling in VDM:

The use of VDM involves the development and analysis of models to help understand systems and predict their properties. Good models exhibit abstraction and rigor. Abstraction is the suppression of detail that is not relevant to the purpose for which a model is constructed.

QUESTION 2: Explain in detail about Sets, Sequences, Maps, Structuring in VDM.

ANSWER:

Sets:

The set type constructor (written set of T where T is a predefined type) constructs the type composed of all finite sets of values drawn from the type T. For example, the type definition. U Group = set of User Id Main Operators on Sets (s, s1, s2 are sets)

 Set enumeration:

the set of elements a, b and c: {a, b, c}.

 Set comprehension:

the set of x from type T such that P(x): {x | x:T & P(x)} The set of integers in the range i to j: {i, ..., j.

Sequences:

The finite sequence type constructor (written seq of T where T is a predefined type) constructs the type composed of all finite lists of values drawn from the type T. For example, the type definition. String = seq of char.

Main Operators on Sequences (s, s1,s2 are sequences)

 Sequence enumeration:

the sequence of elements a, b and c: [a, b, c]

 Sequence comprehension:

sequence of expressions f(x) for each x of (numeric) type T such that P(x) holds (x values taken in numeric order): [f(x) | x:T & P(x)] The head (first element) of s: hd s

Maps:

A finite mapping is a correspondence between two sets, the domain and range, with the domain indexing elements of the range. It is therefore similar to a finite function. The mapping type constructor in VDM-SL (written map T1 to T2 where T1 and T2 are predefined types) constructs the type composed of all finite mappings from sets of T1 values to sets of T2 values. For example, the type definition. Birthdays = map String to Date Main Operators on Mappings

 Mapping enumeration:

a maps to r, b maps to s: {a |-> r, b |-> s} Mapping comprehension: x maps to f(x) for all x for type T such that P(x): {x |-> f(x) | x:T & P(x)}.

 The domain of m:

dom m The range of m: rng m

Structuring:

The main difference between the VDM-SL and VDM++ notations are the way in which structuring is dealt with. In VDM-SL there is a conventional modular extension whereas VDM++ has a traditional object-oriented structuring mechanism with classes and inheritance.

QUESTION 3: The Case Study: Incubator Control • The hardware increments or decrements the temperature of the incubator in response to instructions • Each time a change of one degree has been achieved, the software is informed of the change • According to the safety requirements, the temperature of the incubator must never be allowed to rise above 10 Celsius, nor fall below 10 Celsius. Explain in detail the following.

1. The Class Diagram of Incubator Control in VDM.

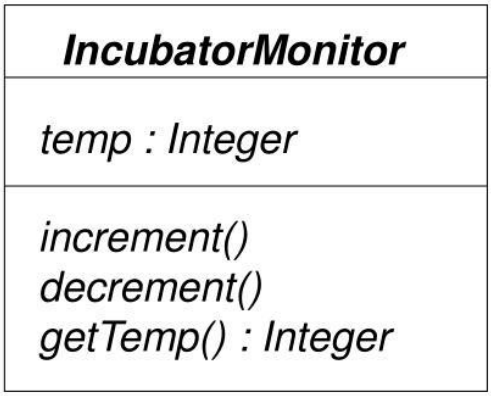
2. Specific the four basic section of operation in VDM.

3. Show the four operations for increment#, decrement# 1.

ANSWER:

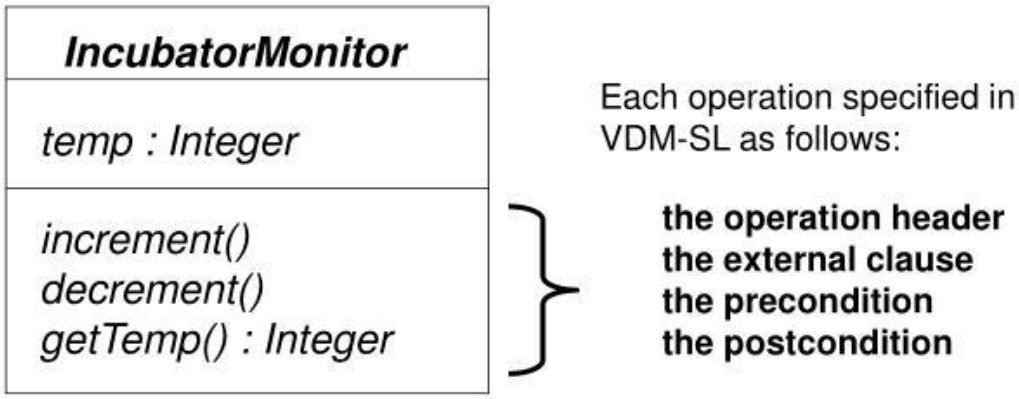
1. The Class Diagram of Incubator Control in VDM:

The UML specification



2. Specific the four basic section of operation in VDM

Specifying the operations in VDM-SL



3. Show the four operations for increment#, decrement# 1.

The *increment* operation

increment()

ext **wr** $temp : \mathbb{Z}$

pre $temp < 10$

post $temp > \overline{temp}$

$\overline{temp} + 1 = temp$

$temp - \overline{temp} = 1$

The *decrement* operation

decrement()

ext **wr** *temp* : ℘

pre *temp* > -10

post $temp = \overline{temp} - 1$