Final Exam Review

Gunter Mussbacher ♦ McGill University
gunter.mussbacher@mcgill.ca

Question 0

A data abstraction which contains procedural abstractions. It encapsulates data and behavior and describes a set of similar objects that have the same structure and behavior.

Question 1

A simple piece of data that exists only when the object exists.

Question 2

Has identity and may be in different states during its lifetime.

Question 3

Specifies how many instances of one class relate to how many instances of another class.

Question 4

A relationship where one instance of a class cannot be related to the same instance of another class more than once for the same relationship. Does not describe a whole-part relationship.

Question 5

Executes two or more operands at the same time.

Question 6

Is triggered by an event as long as its guard is true, and then executes its actions in response.

Question 7

Describes a common design problem, the forces that are at play, and the essence of a common solution.

Question 8

Divides an application into a part that handles the user interface, a part that handles domain concepts, and a part that handles business logic.

Question 9

The collection types used in constraints.

Question 10

Determines the direction of an association.

Question 11

Shared by all instances and typically used for default values, constants, or internal data structures needed by operations.

Question 12

Specifies a predefined list of choices.

Question 13

Describes a whole-part relationship that is transitive and antisymmetric and where the lifecycle of a part is tied to the lifecycle of its parent.

Question 14

Chooses exactly one operand out of potentially many operands based on their guard conditions.

Question 15

Reflects the history of an object up to the current point in time. Describes a mode of an object.

Question 16

Ensures that an object can change roles or possess multiple roles without "changing its class".

Question 17

Goes from a lifeline of an object (the dashed line) to an object (the box).

Question 18

The result of navigating an ordered association / composition / aggregation.

Question 19

Contains an attribute that is related to two classes but cannot be put in either of these classes. An instance of it cannot exist more than once for each pair of these classes.

Question 20

Describes a whole-part relationship that is transitive and antisymmetric and where the lifecycle of a part is not tied to the lifecycle of its parent.

Question 21

May be private, protected, public, or package.

Question 22

The mechanism by which more specific elements incorporate structure and behavior defined by more general elements.

Question 23

Interaction among two objects which may be synchronous or asynchronous.

Question 24

Actions that occur when entering a state, when leaving a state, and while in a state.

Question 25

Simplifies the view that a client has of a complex package.

Question 26

Useful pieces of code are copy-pasted throughout the codebase.

Question 27

The result of navigating at least two unordered associations / compositions / aggregations.

Question 28

The mechanism that allows a subclass to contain a new version of a method instead of inheriting it from a superclass.

Question 29

A property of object oriented software by which an abstract operation may be performed in different ways in different classes.

Question 30

A class that cannot be instantiated.

Question 31

The isa rule, a subclass must retain its distinctiveness throughout its life, and all inherited features must make sense in each subclass.

Question 32

Executes an operand a specific number of times.

Question 33

Hierarchical structuring that allows the number of transitions to be reduced with the help of group transitions, which in turn reduces complexity.

Question 34

Separates possibly multiple representations of object state from the object itself.

Question 35

Immutable, lazy, unique, autounique, const, and defaulted.

Question 36

The result of navigating at least two associations / compositions / aggregations where at least one of them is ordered.

Question 37

Occurs when the decision about which method to run can only be made at runtime.

Question 38

A method that does not have an implementation at that level in the inheritance hierarchy but a concrete method must exist at a lower level.

Question 39

Defines a set of operations that make sense in several classes but cannot have any executable statements.

Question 40

Each object is distinct from each other object, and can be referred to. Two objects (twins) are distinct even if they have the same data.

Question 41

Decides whether a single operand should be executed based on its guard condition.

Question 42

Remembers the last active sub-state before the most recent exit from a composite state.

Question 43

Used to share common parts of an implementation while allowing subclasses to refine other parts by injecting behavior at standard extension points.

Question 44

A large class that is lacking cohesion or makes most of the processing decisions.

Question 45

The result of navigating an unordered association / composition / aggregation.

Question 46

The implementation of an operation.

Question 47

Used to implement an attribute or used to implement an association.

Question 48

Not an object, but rather a reference to an object or no object at all.

Question 49

Determines what classes of objects a variable may contain.

Question 50

Executes one operand based on its guard conditions and, if the guard is true, stops executing the rest of the sequence diagram.

Question 51

A concurrent perspective of a state machine that reacts to shared events, responds simultaneously, and interacts with other perspectives through shared variables or explicit messages.

Question 52

Models a set of related objects that share common information but also differ from each other in important ways.

Question 53

Identifies a message for a static method.

Question 54

Expressed by for all and exists operators in constraints.

Question 55

A specification of a transformation or query that an object may be called to execute.

Question 56

An instance variable or class variable.

Question 57

A list of characteristics that describes a concept.

Question 58

A technique that systematically goes through a textual description to identify classes and attributes.

Question 59

Alt, opt, loop, par, and break in a combined fragment.

Question 60

May contain states, transitions with events and guards, but no actions.

Question 61

Allows clients to treat individual objects and collections of them identically.

Question 62

Models are created at the right level of abstraction using the most appropriate modeling formalism and are transformed to more detailed models.

Question 63

Used for the evaluation of constraints, in contrast to the definition of constraints which is based on classes and their properties.

Question 64

The hiding of implementation details of a class (i.e., variables and methods), typically with the help of an interface that allows only some variables and methods to be seen from the outside.

Question 65

A set of elements that belong to a concept.

Question 66

Visualizes one specific set of instances, i.e., a snapshot of the system as it executes, with the help of only objects, links, and attribute values.

Question 67

Embeds modeling abstractions such as UML attributes, associations, compositions, and state machines directly in a programming language.

Question 68

Represents an entity that participates in an interaction by sending and receiving messages.

Question 69

A transition without a trigger that goes from a composite state to another state.

Question 70

Reduces the need to load into memory large numbers of heavyweight objects from a database or server, when not all of them will be needed.

Question 71

It is suitable to describe systems that react to discrete events but is not suitable to describe continuous systems.

Question 72

The union of two sets without the intersection of the two sets.

Question 73

A taxonomic relationship between a more general and a more specific element. It is transitive and antisymmetric. The more specific element is fully consistent with the more general element and contains additional information.

Question 74

Creating a simplified representation of something for a purpose.

Question 75

A bidirectional relationship between two classes needs to be reflected consistently at both sides of the relationships.

Question 76

The process of making an object out of a concept.

Question 77

The vertical bar on top of a lifeline in a sequence diagram.

Question 78

Saved for later treatment because the current state cannot respond but another state can.

Question 79

Design your system for what you need today, periodically refactor and improve the system design.

Question 80

Defers the responsibility of creating objects to subclasses that know what to create.

Question 81

Can be used in guards of state machines and in constraints.