Alexandria University
Faculty of Engineering
Department of Computer & Systems Engineering
FEB, 2021
Course title Number:
CSE- 223 Programming -2
2^{nd} level
Time allowed: 2 Hours

جامعة الإسكندرية
كلية الهندسة
قسم الحاسب وهندسة النظم
فبراير 2021
اسم المقرر والرقم الكودي له
CSE-223برمجة-2
المستوي الثاني
الزمن: 2 ساعة

الامتحان مكون من 40 سؤال كل منها يساوي درجة واحدة
اجب على جميع الاسئلة بتظليل الدائرة المقابلة لاختيارك في ورقة الاجابة تظليلا كاملا
في حالة تظليل أكثر من اختيار واحد للسؤال تعتبر الاجابة خاطئة - لا تستخدم ظهر ورقة الإجابة أو هامشها في كتابة أي شيء
يمنع منعا باتا استخدام الآلة الحاسبة أو التليفون المحمول أو الأجهزة اللوحية ويلغى امتحان الطالب في هذه الحالة مباشرة

Dynamic Linking library

Disc operatic System

Answer All 40 Questions. Choose between A, B, C, D, and E

1. The filter design pattern
   is the same as the criteria design pattern. ✓
   is choosing among a pool of objects but is never allowed to change them. ✓
   is perfectly nested within calls.
   all of the above
   none of the above

2. The composite design pattern
   builds a tree structure. ✓
   is typically used for representing hierarchies. ✓
   is the same as flyweight.
   (a) and (b).
   (a), (b) and (c).

3. Import modules usually provide some
   read-only interface to prevent changing the important values.
   adaptor interface to match the incoming format with the internal format.
   template to unify the logic of the import.
   a snapshot prototype to provide a consistent object.
   all of the above.

4. The bridge design pattern
   is useful when there is a hierarchy of abstractions and a corresponding hierarchy of implementations. ✓
   forces to reuse logic common to different implementations of an abstraction. ✓
   enables to create a new implementation of an abstraction without having to re-implement the common logic of the abstraction. ✓
   keeps classes that represent an abstraction independent of the classes that supply an implementation for the abstraction.
   (e. All of the above.

5. A fa çade design pattern
   usually has one class that needs proper documentation. ✓
   hides the internals of a package.
   has one class with public methods.
   (a) and (b)

[✓ c.] (a), (b) and (c).

6. The flyweight design pattern

sacrifices memory for speed. ✗

sacrifices speed for memory.

helps model consistency of attributes

(a) and (c). ✗

(b) and (c).

Dynamic [illegible] [illegible]

7. Dynamic linkage is

used with windows DLL files.

allows change in implementation of classes at run time. ✓

a nice tool to configure the system behavior.

all of the above.

none of the above.

8. The decorator can be used to

hide passwords in display.

create simple interface that makes all classes interchangeable.

allow for add-ons at runtime. ✗

all of the above.

none of the above.

9. Cache management is

a prominent creational design pattern ✗

used to sacrifice speed for memory ✗

used to allow requests for all objects of the system to be be more quickly retrieved.

all of the above ✗

none of the above

10. The prototype design pattern

allows the user to experiment with different designs before settling on one.

allows the creation of objects without knowing their exact way of creation. ✓

is essential in the rapid prototyping paradigm.

does not need shallow or deep copying techniques.

none of the above.

11. The prototype design pattern

can dynamically add prototypical objects at runtime.

ensures that the created objects provide a consistent set of methods for the client object to use.

needs additional time in order to write the PrototypeBuilder.

(a) and (c).

(a), (b) and (c).

12. The Singleton

is the same as the read-only interface.

is the same as the snapshot.

must take care of the concurrency while calling its getInstance() method.

must take care of the concurrency while calling its constructor method.

hard to detect in existing code.

13. The object pool design pattern

contains a singleton. ✓

is a creational design pattern. ✓

is mainly used in structural design patterns.

(a) and (b).

(b) and (c).

14. The soft references in the object pool

must be implemented carefully.
are more flexible than hard references.
limit the size of the pool.
ensures a maximum number of instances.
all of the above.

15. The relationship between an invoice and the lines in an invoice is:

One-to-one.
One-to-many.
Many-to-many.
No relation.
None of the above.

16. Java Package containing lots of abstract classes should

change regularly.
not change often.
contain lots of concrete classes.
contain all concrete classes depending on the abstract classes in the same package.

17. The following holds in a mutli-threaded program:

A thread can start on any object that implements the Runnable interface.
The Runnable interface has only single method called "run" which is the start code of the associated thread.
A thread can be either in state: running, or sleeping, or suspended, or *stopped,* or busy
All the above is true
Some but not all of the above hold

18. In the producer consumer design pattern, the buffer can be implemented as

Queue
Stack
Hashmap
(a) or (b)
(a) or (c)

19. How much info about the change should the subject send to the observers in the observer design pattern?

Push Model – Lots ✓
Push Model – Very Little
Pull Model - Very Little
(a) and (c)
(b) and (c)

20. In the cache management design pattern, which object is removed from the cache?

Least recently used. ✓
Least frequently used.
Most recently used.
Configurable (a) or (b).
Configurable (b) or (c).

21. The chain of responsibility design pattern is used

when more than one object may handle a request and the actual handler is known in advance.
when more than one object may handle a request and the actual handler is not known in advance.
when the requests are always handled where they are generated.

when we want to couple the sender and the receiver of a request.
only when the chain of handlers is static.

22. The chain of responsibility design pattern is used

in the JAVA AWT 1.0
in JAVA 5 after the introduction of generics
in java.util.* as part of the Java stack
in the Spring boot
nowhere

23. the command design can be used in

implementing the undo functionality ✓
implementing menu and toolbar functionality
implementing a functor
all of the above
some of the above

24. We can use the Little Language design pattern

in implementing a DOS console ✓
in parsing SQL statements ✓
to evaluate a logical expression
all of the above
some of the above

25. We can use the Little Language objects

in a declarative way
in a chained way
in a concatenated way
in an isolated way
in a concurrent way

26. The mediator design pattern is

needed to reduce communication complexity. ✓
needed to tightly couple negotiators. ✗
not allowed to determine priorities between competing objects over one resource.
part of the structural design patterns.
none of the above.

27. The snapshot design pattern cannot be used

to save the state of the objects before pressing the cancel button ✗
to capture the version of a word document in case of a system crash
to coexist with a caretaker object ✗
to document the readings of a group of sensors ✗
to save a picture in jpg format.

28. The observer designer pattern

is equally efficient in capturing UI events compared to the chain of responsibility. ✗
is less efficient in capturing UI events than the chain of responsibility.
typically has an observe method.
provides an interface for attaching and detaching Observer objects
none of the above

29. Observers

cannot be added without modifying the subject ✗
can be added/removed at any time ✓
cannot reuse subjects without reusing their observers ✗
can keep track of their subjects using a stack. ✗

cannot subscribe to specific events of interest ✗

30. How can an observer be notified only after several subjects have changed state?
By chaining Observers
By running them in concurrent threads
By using an intermediary object which acts as a mediator ✓
By implementing a simple ORing function within the observer.
By using a functor

31. In the state design pattern,
an object per state is created. ✓
one object is created with array of states as a private variable. ✗
all state objects must be created at the beginning. ✓ ?
a state-transition table is defined as static. ✗
an object is allowed to have multiple states at the same time. ✗

32. An example of strategy pattern is,
sorting in a grid by arbitrary columns. ✓
validating input of text fields. ✓
print using PDF, laser printer, jpg.
all of the above.
some of the above.

33. In the visitor design pattern,
we must have a complex data structure to visit its nodes. ✓
the node must accept the visitor object. ✓
the visitor object is actually executing the required logic. ✓
all of the above.
some of the above.

34. The benefits of the visitors include
adding new operations is easy.
related behavior is evenly spread over the classes defining the data structure. ✓
visitors are prohibited from accumulating state; which simplifies complexity.
adding new ConcreteElement classes is easy
none of the above.

35. Visitor and composite design patterns
contradict each other.
combine easily. ✓
form together the momento design pattern.
are part of the family of structural design patterns.
can be used interchangeably.

36. The method Thread.currentThread()
is static.
returns a handle for the current thread.
creates a new thread.
(a) and (b).
(a) and (c).

37. Which statement is true?
wait() forces the current thread to give up the monitor and go to sleep.
notify() wakes up the first thread that called wait on the same object.
notifyAll() wakes up all threads that are waiting on the same object.
All the above.
Some of the above.

38. Which statement is true?

[✓ Balking is similar to guarded suspension but balking returns rather than waiting.]
Balking is similar to guarded suspension but guarded suspension returns rather than waiting.
Balking wastes resources more than guarded suspension.
Guarded suspension wastes resources more than balking.
None of the above.

39. In the scheduler design pattern,

we need enter(), done(), and schedule() methods.
the processor does not perform the computation abstracted by a request.
[✓ we need a variable to be shared by the enter and done methods.]
we do not need any local variables since the synchronized keyword solves all synchronization issues between threads.
None of the above.

40. Which is a special form of a Scheduler?

Balking
Read/Write Lock
Two-Phase Termination
Producer-Consumer
Guarded Suspension

<u>*END OF EXAM*</u>

K. Nagi