

Welcome

Help us design a safer smart contract language for writing blockchain programs!

If you are an experienced object-oriented programmer and are at least 18 years old, we are interested in your help with a research study about programming languages. In the study, you will help us learn how to design languages to make software engineers more effective. The study takes about four hours to complete on the (anonymized) campus; you will be offered a \$75 Amazon gift certificate for completing it. To apply to participate in the study, please complete this brief questionnaire.

Informed consent

This survey is part of a research study conducted by (anonymized) at (anonymized) University and is funded by the (anonymized).

Summary This task involves several programming questions. We will use your answers to the questions to help guide the design of a new programming language that is intended to make it easier for programmers to write software effectively.

Purpose The purpose of the study is to inform and evaluate the design of a

programming language for blockchain computing platforms.

Procedures

In this study, you will be asked to complete programming-related tasks, such as writing code, reading code, or debugging.

Participant Requirements

Participation in this study is limited to individuals age 18 and older located in the United States who have at least a year of Java experience.

Risks

The risks and discomfort associated with participation in this study are no greater than those ordinarily encountered in daily life or during other online activities.

Benefits

There may be no personal benefit from your participation in the study but the knowledge received may be of value to humanity.

Compensation & Costs

If you are completing this experiment as a freelance worker, you will be paid the amount advertised for this task if you complete it.

There will be no cost to you if you participate in this study, but you are responsible for your own Internet access, computer, etc.

Future Use of Information

In the future, once we have removed all identifiable information from your data (information or bio-specimens), we may use the data for our future research studies, or we may distribute the data to other researchers for their research studies. We would do this without getting additional informed consent from you (or your legally authorized representative). Sharing of data with other researchers will only be done in such a manner that you will not be identified.

Confidentiality

If you are a Freelancer.com participant, the data captured may include your public profile information, such as your username.

The data may also include your IP address. We will not ask for any other personally-identifiable information.

By participating in this research, you understand and agree that (anonymized) may be required to disclose your consent form, data and other personally identifiable information as required by law, regulation, subpoena or court order. Otherwise, your confidentiality will be maintained in the following manner:

Your data and consent form will be kept separate. Your consent form will be stored in a secure location on (anonymized) property and will not be disclosed to third parties. By participating, you understand and agree that the data and information gathered during this study may be used by (anonymized) and published and/or disclosed by (anonymized) to others outside of (anonymized). However, any personally identifiable information, including your name, will not be mentioned in any such publication or dissemination of the research data and/or results by (anonymized). We may publish location information (inferred from IP addresses) in aggregate form. Note that per regulation all research data must be kept for a minimum of 3 years. Research sponsors, including the US Department of Defense, may have access to research records.

Right to Ask Questions & Contact Information

If you have any questions about this study, you should feel free to ask them by contacting the Principal Investigator now: (anonymized). If you have questions later, desire additional information, or wish to withdraw your participation please contact the Principal Investigator by mail or e-mail in accordance with the contact information listed above.

If you have questions pertaining to your rights as a research participant; or to report concerns to this study, you should contact the Office of Research integrity and

Compliance at (anonymized) University. Email: [\(anonymized\)](#) . Phone: (anonymized).

Voluntary Participation

Your participation in this research is voluntary. You may discontinue participation at any time during the research activity. You may print a copy of this consent form for your records.

I am age 18 or older.

Yes

No

I am located in the United States.

Yes

No

I have read and understand the information above.

Yes

No

I want to participate in this research and continue with the survey.

Yes

No

Default Question Block

This screening test evaluates your Java and programming knowledge. You do not need to answer all questions correctly in order to participate; please just do the best you can.

Which of the following might be a valid Java constructor invocation?

Square.new(5)

malloc(sizeof(Square))

square(5)

new Square(5)

In Java, *encapsulation* refers to:

Serializing data correctly so that it is transmitted properly between systems

Preventing clients from improperly depending on implementation details

Using the `capsule` keyword to protect secret data

```
void test() {  
    ArrayList list1 = new ArrayList();  
    list1.add(1);  
  
    ArrayList list2 = list1;  
    list2.add(2);  
  
    System.out.println(list1.size());  
}
```

If `test()` is run, what is the output?

1

2

Do not use any external resources to answer this question.

Which statements are true of interfaces in standard Java?

	True	False
Methods in interfaces (except for <code>default</code> methods) lack bodies.	<input type="radio"/>	<input type="radio"/>
Methods in interfaces are public by default.	<input type="radio"/>	<input type="radio"/>
A class can implement no more than one interface.	<input type="radio"/>	<input type="radio"/>
Interfaces have no field declarations unless they are <code>public static final</code> .	<input type="radio"/>	<input type="radio"/>

Do not use any external resources to answer this question.

Which statements are true of `static` methods in standard Java?

	True	False
They are not implemented where they are defined, but must be overridden in a subclass.	<input type="radio"/>	<input type="radio"/>
They cannot be	<input type="radio"/>	<input type="radio"/>

overridden.

They can be invoked without having an instance of the class.

These page timer metrics will not be displayed to the recipient.

First Click: *0 seconds*

Last Click: *0 seconds*

Page Submit: *0 seconds*

Click Count: *0 clicks*

Do not use any external resources to answer this question.

```
public abstract class Customer {  
    private String name;  
    public Customer (String name) {  
        this.name = name;  
    }  
    public String getName() { return name; }  
    public abstract void buy();  
}
```

Which statements are true about `Customer`?

True

False

The `Customer` class cannot be extended.

The `Customer` class cannot be instantiated.

Concrete subclasses must

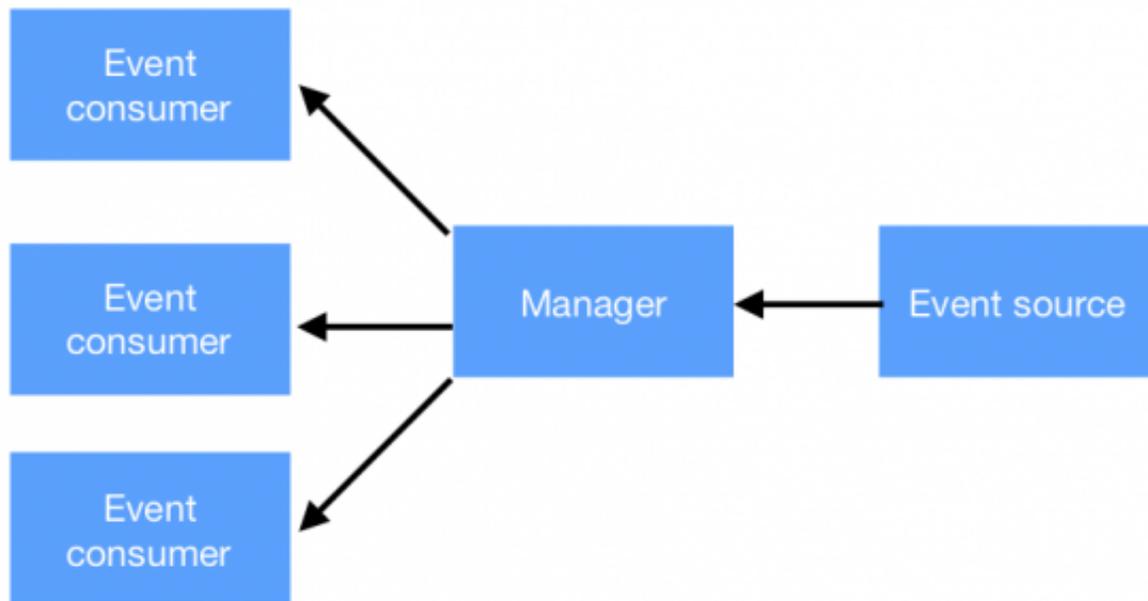
implement the
`buy()` method.

Subclasses must
implement the
`buy()` method.

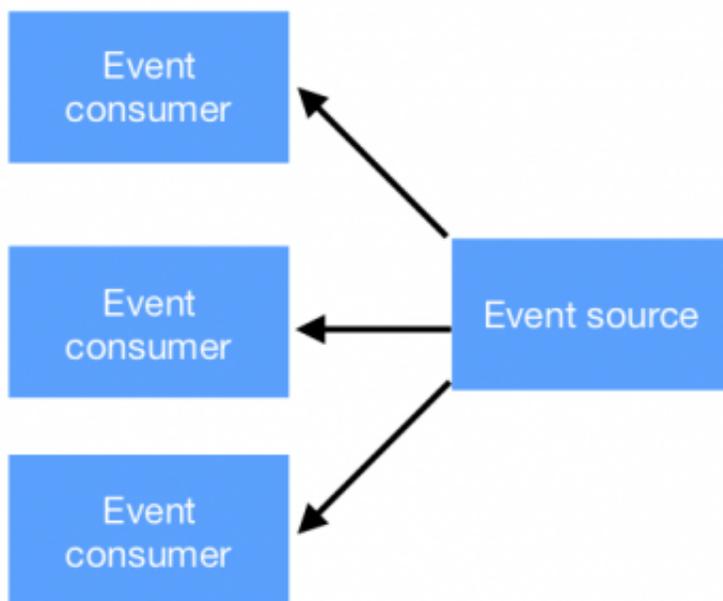


Block 1

Design A



Design B



A software engineer is choosing between designs A and B for a component in which consumers need to find out when certain events occur. Which consideration would best justify choosing Design A over Design B?

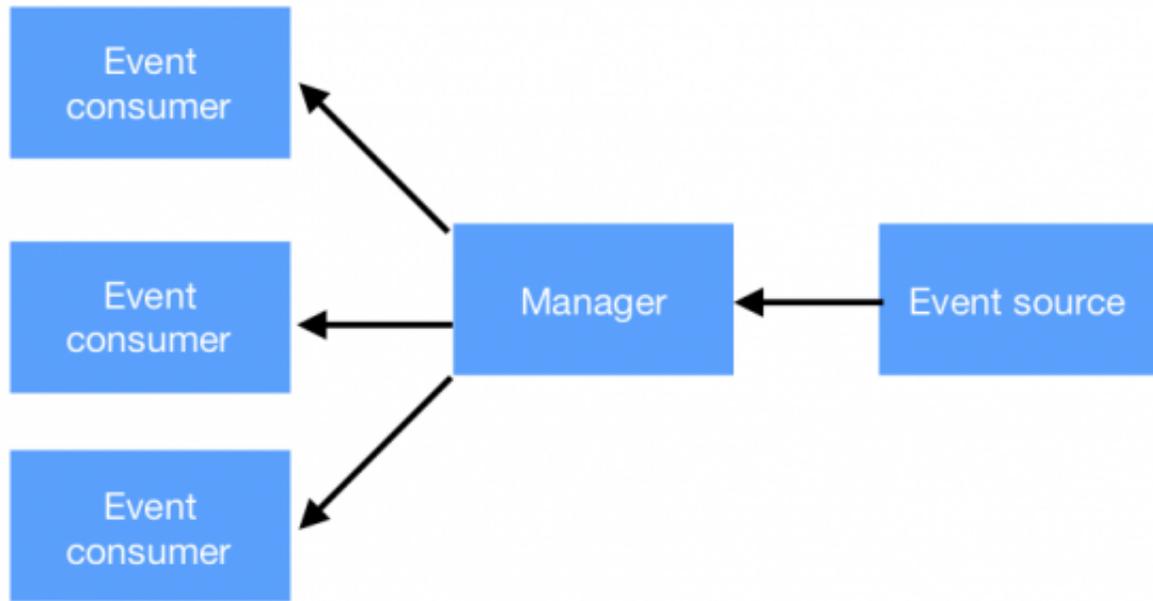
The event source should be decoupled from the list of event consumers so that the source need not be aware of the consumers.

None; Design B is better because it is simpler.

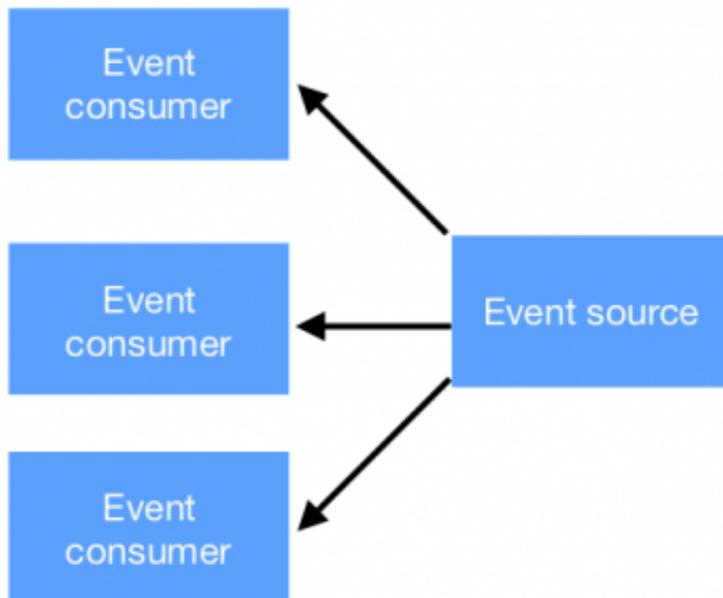
Many different sources may need to send the event to different sets of consumers.

Event sources may be implemented by third parties.

Design A



Design B



Again considering the above diagram, what is Design A typically called?

Manage-events

Publish-subscribe

Source-consume

A pattern in which there can be only one instance of a given class is called:

Flyweight pattern

Singleton pattern

Unique pattern

Builder pattern

Thanks

Thank you! If you are eligible to participate in the study, you will receive an email.

Please enter your email address:

Powered by Qualtrics