dormir2.log

acceso

módulo cpp 05

dormir2 · 22 de noviembre de 2021

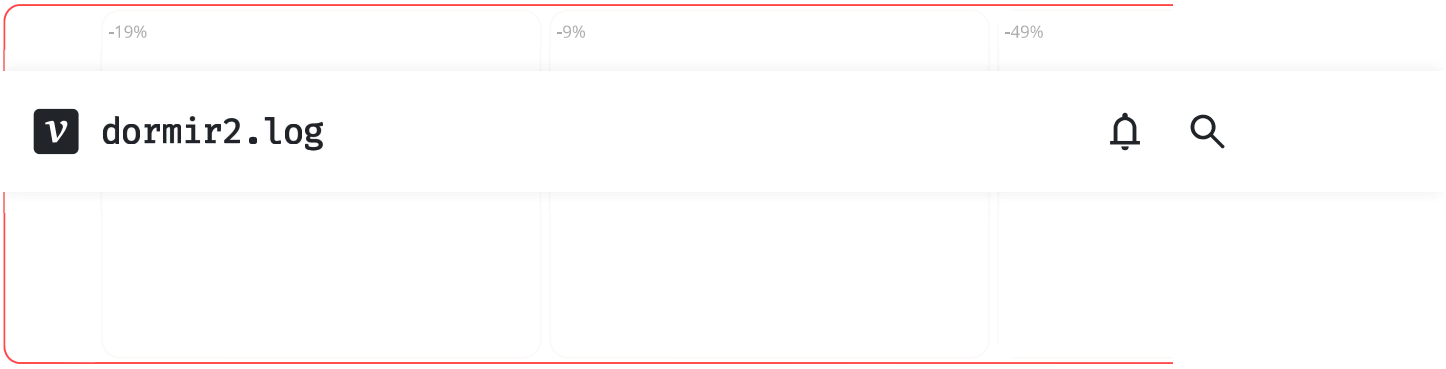
seguir 0

cpp

42Seúl

Vista de la lista 14 / 15

dormir2.log



ex00

Crea una clase burócrata.

- Crear un nombre y grado
- Si la calificación es mayor que 1 (si es menor que 1),
  se lanza Bureaucrat::GradeTooHighException.
- Si la calificación es menor que 150 (mayor que 150), implemente
  Bureaucrat::GradeTooLowException.
- Crear (getName y getCalificación)

Anule el << de ostream para generar como se muestra a continuación.

> < nombre >, grado burocrático < grado >

Excepción personalizada

```
class exception {
public:
  exception () throw();
  exception (const exception&) throw();
  exception& operator= (const exception&) throw();
  virtual ~exception() throw();
  virtual const char* what() const throw();
}
```

Estos son datos plusplus. Aquí, what() es un método virtual, así que impleméntelo anulando este método.

dormir2.log

ex01

Crear una clase de formulario

- nombre, grado, sign_grade, signo
- Si grade o sign_grade no está entre 1 y 150,
  se lanza HighException o LowException como antes.

- Implementar el método beSigned
  Obtener la calificación de Burócrata y si es superior a sign_grade (en la materia, si la calificación es lo suficientemente alta, dice) (calificación <= sign_grade)

clase burocrática

- Implemente el método signForm.
  Este método analiza el valor obtenido mediante beSgined y hace lo siguiente:

> Éxito - < burócrata > firma < formulario >
> Fracaso - < burócrata > no puede firmar < formulario > porque < motivo >

Esto se puede hacer rápidamente si usas bien la excepción personalizada, ¿verdad?

ex02

Árbol ASCII: https://ascii.co.uk/art/tree
o: https://asciiart.website/index.php?art=plants/trees

Clase ShrubberyCreationForm

- Es privado y tiene el nombre target.
- Firme 145, se requiere exec 137.
  Si la ejecución es exitosa,
  cree un archivo con <destino>_shrubbery y guarde el árbol ASCII.

Formulario de solicitud de robótica

- Es privado y tiene el nombre target.
- Se requiere el signo 72, el ejecutivo 45.

dormir2.log

Formulario De Indulto Presidencial

- Es privado y tiene el nombre target.
- Firma 25, se requiere ejecutivo 5.
  Si la ejecución se realiza correctamente, se mostrará el siguiente mensaje.
  < target > ha sido indultado por Zafod Beeblebrox.

Las tres clases anteriores heredan la clase Formulario.
Haga que el método de ejecución en el formulario sea un método virtual y luego impleméntelo en

cada clase.

El método de ejecución recibe Burócrata, verifica si es un signo e implementa cada una de las acciones anteriores si el signo es verdadero y la puntuación es suficiente. De lo contrario, imprima un mensaje de error.

Se convierte en una batalla con el casting. Esto se puede resolver mediante el uso de conversión, como usar un tipo no constante como constante, o viceversa.

ex03

Crear una clase de pasante

- El método makeForm
  recibe dos cadenas como parámetros.
  Uno es sobre qué forma utilizar y el otro es el objetivo.

Implemente un método que ejecute el formulario de manera diferente según su contenido y devuelva un puntero. Sin embargo, no abuses de if/else if.

Esto se puede resolver usando una declaración de cambio. Si ha realizado el módulo 01 de cpp, puede hacerlo fácilmente.

seguir

dormir2

dormir2.log

Publicación anterior
módulo cpp 04

0 comentarios

Escribir un comentario

Escribir un comentario

관심 있을 만한 포스트

Java 오버라이딩과 가상 메서드

오버라이딩, 정적 바인딩과 동적 바인딩, C++에서의 가상 함수, Java 에서의 가상 메서드

2021년 6월 2일 · 1개의 댓글

by **Codren** 2

C++ 면접 질문 정리

dormir2.log

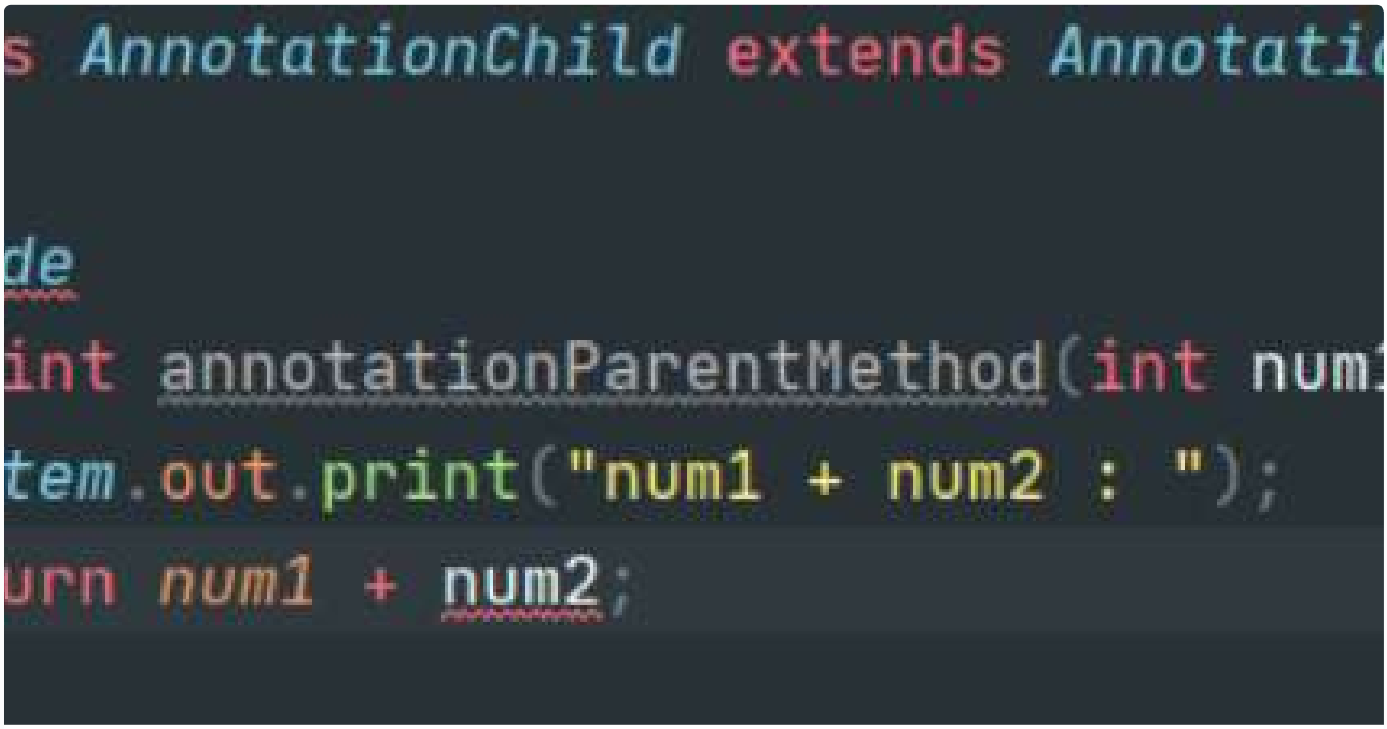
by **초보개발** 10

[면접 준비] Swift + RxSwift

iOS 개발자 면접을 위한 개념 정리 - Swift / RxSwift

2023년 4월 14일 · 0개의 댓글

by **김상우** 5



@Override 어노테이션의 의미와 사용 이유는 무엇일까?

어노테이션은 JDK5 부터 등장하였으며, 클래스나 메서드, 변수에 @을 사용하는 것을 말합니다.어노테이션은 사전적 의미로 주석을 뜻합니다. 주석과는 역할이 다르지만, 주석처럼 달아 특수한 의미 부여가 가능하며, 기능 주입이 가능합니다. 어노테이션을 사용하는 가장 큰 이...

2021년 5월 16일 · 0개의 댓글

by **Doa Choi** 6

dormir2.log

TypeScript 클래스와 인터페이스

객체지향 프로그래밍(OOP, Object-Oriented Programming)은 애플리케이션을 개발할 때 코드 중복을 획기적으로 줄일 수 있는 방법이다. 객체지향 프로그래밍은 커다란 문제를 클래스라는 단위로 나누고 클래스 간의 관계를 추가하면서 코드 중복을 최소화하는...

2020년 4월 19일 · 1개의 댓글

by 김대현 ❤ 5

DRF 공부하기 (7) :: GenericAPIView와 Mixins

https://velog.io/@jcinsh/DRF-6-GenericView%EC%99%80-Mixin와 https://ssungkang.tistory.com/entry/Django-APIView-Mixins-generics-APIView-ViewSet%EC%9D%84...

2020년 10월 18일 · 1개의 댓글

by PHYYOU ❤ 3

C++ #07 다형성

07. 다형성 다형성의 기법 Polymorphism이란 여러 개의 서로 다른 객체가 동일한 기능을 서로 다른 방법으로 처리할 수 있는 기능을 의미한다. 예를 들어 칼, 대포, 총 등의 무기들은 공통적으로 '공격'이라는 동일한 기능을 다르게 수행할 수 있다. image.png 따라서 무기 객체에서 attack() 함수를 실질적으로 구현할 필요없이 추상...

2020년 2월 5일 · 0개의 댓글

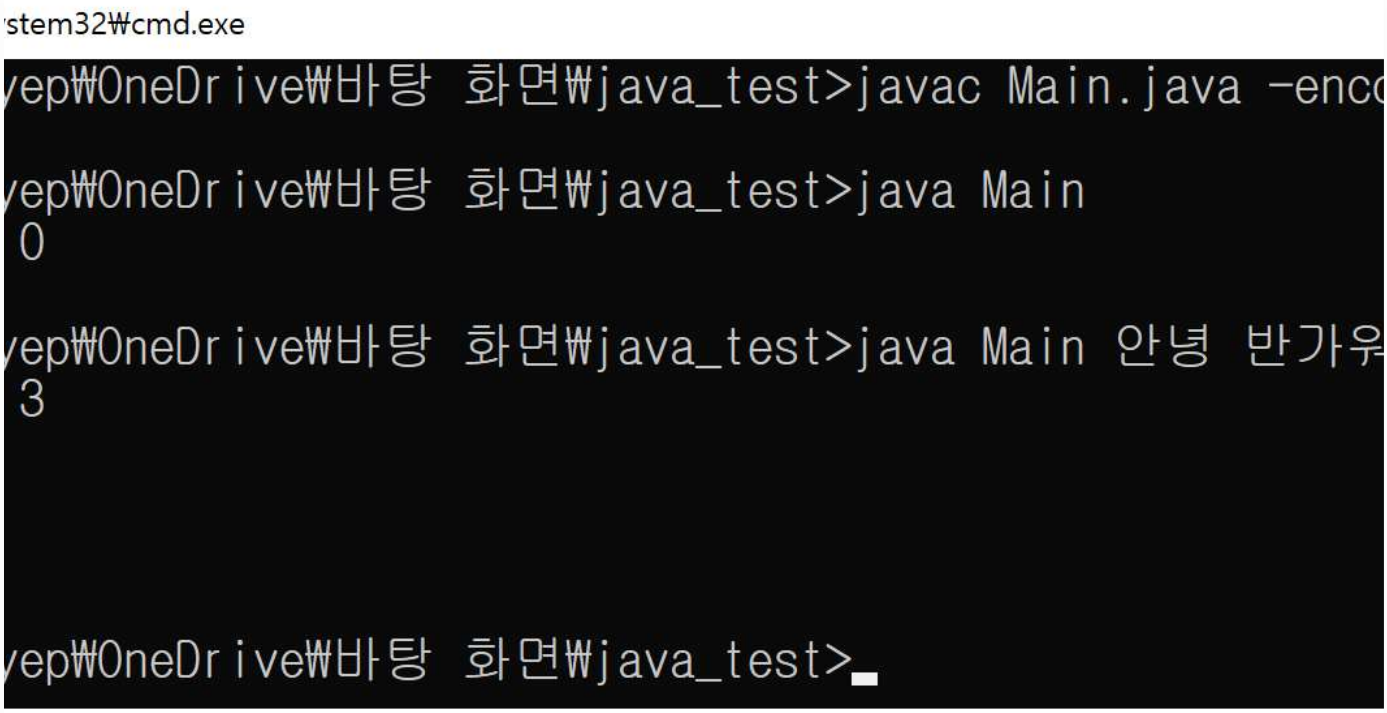
by underlier12 ❤ 1

백엔드 개발 기술면접 정리 (Java 추가중)

안녕 난 그냥 그래 : OOP는 현실 세계를 프로그래밍으로 옮겨와 현실 세계의 사물들을 객체로 보고, 그 객체로부터 개발하고자 하는 특징과 기능을 뽑아와 프로그래밍하는 기법입니다. : 장점 코드에 대한 재사용이 용이하다. 유지보수에 용이하다.객체 단위로 코드가 나눠...

2021년 10월 10일 · 1개의 댓글

by **희소** 12

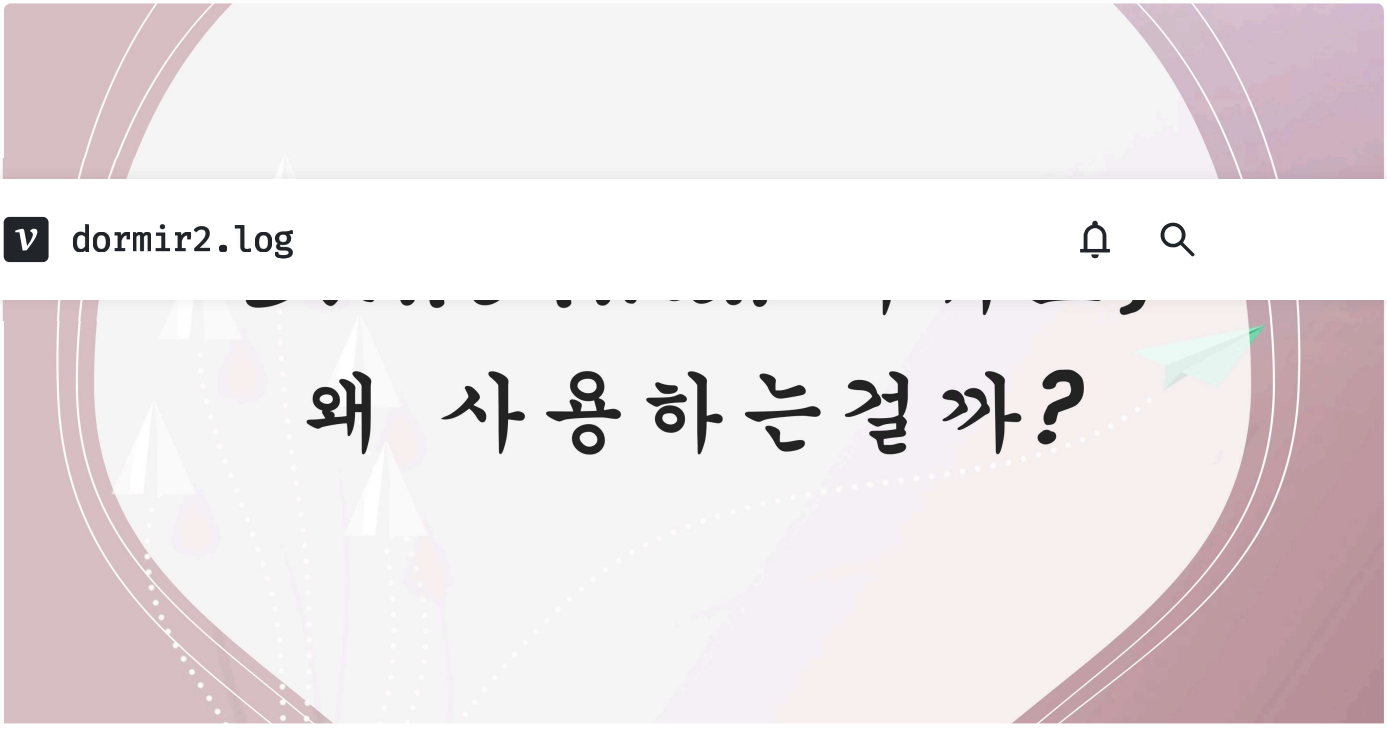


[JAVA] 메인 메소드 public static void main(String[] args) 알아보기

메인 메소드 public static void main(String[] args) 에 대해 알아봅시다.

2021년 4월 13일 · 1개의 댓글

by **skyepodium** 7



[Swift] final 키워드, 왜 사용하는걸까?

Swift에서 final 키워드를 사용하는 의미는 무엇이고, 적용하면 어떤 이점이 있을까요?

2021년 5월 1일 · 0개의 댓글

by **Ryan (Geonhee) Son**

♥ 1

Powered by Stellate