

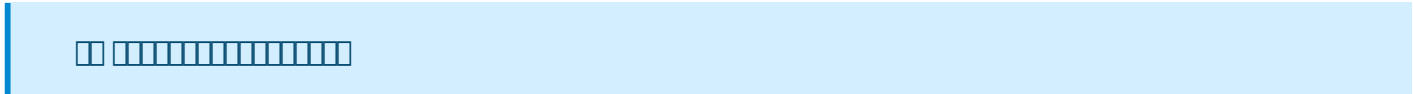
Java

□□□□□

□□□□□□

- [Java □□ - final □□](#)
- [JDA □□ Discord Forum Channel □□ Thread □□](#)
- [Java □□ - Type Casting](#)

Java 面试题 - final 面试题



final 在 Java 中有什么作用？Code 是什么？

final class 是什么？

final class 是什么？

编译错误？



Override？

编译错误？



编译错误？

final 有什么作用？



- final 和 abstract 不能同时使用
- Reflection API 可以反射 final 成员

JDA ❷ Discord Forum Channel Thread ❷

Code ❷

```
public class Main extends ListenerAdapter {

    public static final String TOKEN = "<BOT_TOKEN_HERE>";
    public static final String SERVER_ID = "<SERVER_ID_HERE>";
    public static final String PROJECT_FORUM_ID = "<FORUM_ID_HERE>";

    public static void main(String[] args) throws InterruptedException {
        JDA jda = JDABuilder.createDefault(Main.TOKEN).build();

        jda.addEventListener(new Main());
    }

    @Override
    public void onReady(ReadyEvent event) {
        System.out.println("Bot is ready");
        JDA jda = event.getJDA();

        Guild server = jda.getGuildById(SERVER_ID);
        assert server != null;
        ForumChannel channel = server.getForumChannelById(PROJECT_FORUM_ID);
        assert channel != null;
        System.out.printf("Channel name: %s%n", channel.getName());
        List<ThreadChannel> threadChannels = channel.getThreadChannels();

        for(ThreadChannel thread : threadChannels){
            System.out.printf("Thread: %s%n", thread.getName());
        }
    }
}
```

```
//  Bot
//  jda.shutdown();
}
}
```



- JDA Guild (Discord Server) Guild ForumChnnel
- `getThreadChannels()` ThreadChannel List
- List for



- **TOKENChannel ID (Hardcode)**
- JDA (JDA.shutdown()) Gradle

Java ☐☐ - Type Casting

Type Casting ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Casting ☐ ☐ ☐

--	--	--	--	--	--	--	--

```
class Animal {
    void speak(){
        System.out.println("Default speak");
    }
}
```

```
class Bird extends Animal {
    @Override
    void speak() {
        System.out.println("☐☐");
    }
}
```

```
class Dog extends Animal {
    @Override
    void speak() {
        System.out.println("🐶");
    }
}
```

```
public class main {  
    public static void main(String[] args) {  
        Animal animal = new Animal();  
        Bird bird = new Bird();  
        Dog dog = new Dog();  
  
        animal.speak();  
        bird.speak();  
        dog.speak();  
    }  
}
```

```
// Upcasting and Downcasting
```

```
Animal animal2;
animal2 = bird; // Upcasting
animal2.speak(); // works normally
```

```
Bird bird2;
// bird2 = animal; // Down casing, compile error
// bird2 = (Bird)animal; // Compile pass
// bird2.speak(); // but throw ClassCastException error
```

```
Dog dog2;
// dog2 = bird; // Obviously not working for compiling
// dog2 = (Dog)bird; // Cannot casting to another child class, compile error.
}
}
```

- Bird → casting → Animal → Upcasting → Upcasting →
- → animal casting → Bird → Animal → Bird → Compiler
 - → Downcasting → Compiler
- → (Bird)animal → Compiler → Compiler → Compiler → OK →
 - → ClassCastException → Runtime error → VM
- → Animal → class → Compiler → Compiler → ...