

LET US LEARN

GENERICS IN

JAVA



ROHIT DOSHI
@ROHITDOSHI9



ROHIT DOSHI
@ROHITDOSHI9

What are Generics?

Uncover the power of generics in Java programming and how they enhance type safety and reusability.

Generics enable the creation of classes, interfaces, and methods that operate with different data types, while maintaining type safety.

They're particularly useful in collection classes.



ROHIT DOSHI
@ROHITDOSHI9

Problem of Type Casting

Without generics, collections can only store objects of type Object. This requires explicit type casting when retrieving elements, leading to runtime errors if not done correctly.

Generics allow you to specify the type of elements a collection can hold at compile time. This eliminates the need for manual type casting and catches type errors early.



ROHIT DOSHI
@ROHITDOSHI9

Syntax : Generic Classes

Declare a generic class using the format :

```
class ClassName<T> { /*...*/ }
```

T is a type parameter that represents the actual type that will be used when an instance of the class is created.



ROHIT DOSHI
@ROHITDOSHI9

Syntax : Generic Methods

Similar to generic classes, methods can also be made generic using the format :

```
public <T> void methodName(T parameter)  
{ /*...*/ }
```

The type parameter T is specified before the return type.



ROHIT DOSHI
@ROHITDOSHI9

Wildcards in Generics

Wildcards like "?" allow greater flexibility in working with generic types.

"?" can represent an unknown type, or it can have specific bounds, such as
"? extends Number"



ROHIT DOSHI
@ROHITDOSHI9

Bounded Type Params

Bounded type parameters restrict the acceptable types to a specific range.

For example, `<T extends Number>` ensures that T can only be a subtype of Number.



ROHIT DOSHI
@ROHITDOSHI9

Benefits of Generics

- Improves code quality
- Catches type-related errors at compile time
- Enhances re-usability
- Makes code more readable
- Doesn't require explicit type casting





ROHIT DOSHI
@ROHITDOSHI9

Generics in Java offer a powerful tool for creating flexible and type-safe code, promoting reusability and reducing errors.

Dive deeper into Java generics by exploring advanced topics like wildcard captures, generic interfaces, and more.





ROHIT DOSHI
@ROHITDOSHI9

WAS THIS USEFUL?



LIKE



COMMENT



REPOST

P.S. Let me know the next topic you would like me to cover!

