21. Třídy a objekty – dědičnost (Java)

**Dědičnost** (Inheritance) je jedna ze základních vlastností OOP a umožňuje vytvářet nové třídy na základě již existujících tříd. Nová třída zdědí atributy (vlastnosti) a metody své rodičovské třídy. To usnadňuje opětovné využití kódu.

K dědění se používá klíčové slovo extends.

Rodič – třída ze které se dědí

Potomek – třída která dědí

Dělení:

1. Přímá/Jednoduchá (Single):

Potomek zdědí všechny atributy a metody pouze od jediné rodičovské třídy. Obsah obrázku text, Písmo, snímek obrazovky, diagram

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

1. Vícenásobná (Multiple Inheritance):

Třída může zdědit atributy a metody z více než jedné rodičovské třídy.

U tříd tento způsob dědění java **nepodporuje,** jelikož může nastat tzv. "diamond problem". U rozhraní je možnost dědit více než jedno rozhraní.

Diamond problem:

Nazývá se tak protože dědění tříd, protože tvoří “diamant“.Obsah obrázku snímek obrazovky, diagram, text, řada/pruh

Popis byl vytvořen automaticky

Třídy **B** a **C** dědí ze třídy **A**. Třída **D** dědí z **B** a **C**.

Pokud existuje metoda nebo atribut ve třídě **A** a obě třídy **B** a **C** tuto metodu přepíší nebo poskytnou vlastní implementaci, stává se nejasné, která implementace by měla být použita ve třídě **D**.

Př. ToString():

Po zavolání ToString()

Třída A vrací “jsem A“

Třídy B a C jí přepíšou a vrací “jsem B“ a “jsem C“

Kterou z těchto implementací si má třída D vybrat “jsem B“ nebo “jsem C“? **=> “diamond problem“**

**Proto Java nepodporuje tento způsob dědění.**

1. Víceúrovňová/Opakovaná (Multilevel):

Potomek(**C**) dědí z rodiče(**B**), který má dalšího vlastního rodiče(**A**). Jelikož jeho rodič(**B**) už podědil atributy od svého rodiče (**A**), získá tak atributy a metody obou tříd.Obsah obrázku diagram, řada/pruh, snímek obrazovky, design

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Super

V jazyce Java klíčové slovo **super** slouží k odkazování na nadřazený (rodičovský) objekt nebo třídu.

Využití:

1. Volání konstruktoru rodiče:

Jeden z hlavních účelů **super** je volání konstruktoru rodiče ze strany potomka. Používá se v konstruktoru potomka na volání konstruktoru rodiče a zajistí, že inicializační kód rodičovské třídy je spuštěn před inicializačním kódem potomka.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

1. Odkaz na metody nebo atributy rodiče:

Může být také použito k odkazování na metody nebo atributy rodičovské třídy, a to zejména v případě, kdy jsou v potomkovi překryté nebo přepsané.

Final

V jazyce Java klíčové slovo **final** má několik významů, a to v závislosti na kontextu, ve kterém je použito.

1. Pro třídu:

Když je třída označena jako **final**, nemůže být děděna (nelze z ní vytvořit potomka).Obsah obrázku text, Písmo, snímek obrazovky, design

Popis byl vytvořen automaticky

1. Pro metodu:

Když je metoda označena jako **final**, nelze ji přepsat (override) v potomkovské třídě.

Obsah obrázku text, Písmo, snímek obrazovky, Grafika

Popis byl vytvořen automaticky

1. Pro atribut

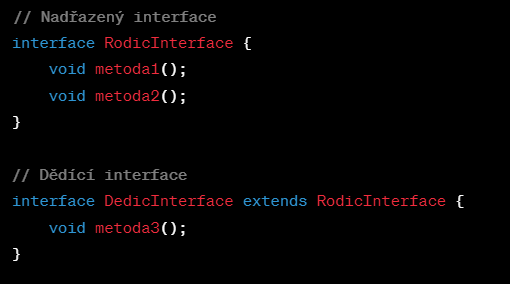
Když je atribut označen jako **final**, nelze mu měnit hodnotu po jeho inicializaci.



Rozhraní

Také rozhraní podporují dědičnost. Dědí metody a konstanty z jiných rozhraní.

**Více informací o dědění rozhraní v materiálu o rozhraních.**



**DedicInterface** bude obsahovat všechny tři metody.

Abstraktní třídy

**// podle Kubinové to patří do rozhraní, ale nechám to i tady**

Je třída, která nemůže být přímo instanciována (nemůžete vytvořit objekt této třídy) a slouží k definování obecných vlastností a metod pro její potomky.

Abstraktní třída je deklarujeme klíčovým slovem **abstract**

Může obsahovat jak konkrétní metody s implementací, tak i abstraktní metody, které jsou deklarovány, ale nemají definovanou implementaci. Potomci abstraktní třídy musí implementovat všechny její abstraktní metody.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky