

# Potíže s dědičností

## Table of Contents

Vtip .....	1
Přínosy dědičnosti .....	1
Hlavní omezení .....	1

## Vtip

Dědičnost. Nejlepší objektově-orientovaný způsob, jak se pohádat a naštvat ty před námi i po nás.

## Přínosy dědičnosti

- Je fajn, *ušetříme psaní kódu* i jeho *objem* (kód je tam jednou), omezí se redundance.
- Někdy dokonce i *předejdeme chybám* tím, že chybu opravíme na jednom místě a OK, protože efekt se hned bez dalšího promítne do všech podtříd = je podděděn.
- Můžeme využít *polymorfismus*, kdy jsme schopni jednotně obsloužit více typů objektů
- například `Employee`, `Manager` nebo `Student` jsou všechno osoby `Person` a mají tedy jméno `getName()`

## Hlavní omezení

- Dědičnost je příklad *silné (těsné, pevné) závislosti*.
- Změna v rodičovské třídě (nadtřídě) má silný vliv na funkci, případně i kompilovatelnosti podtříd.
- Metodu, která se dá překrýt v podtřídě, nesmíme volat z konstruktoru!
- Obecně: z konstruktoru pokud možno nevolat nic.