



Programovanie v jazyku C#

Abstraktné triedy, rozhrania, hierarchia tried

prednáška 4
Ing. Ján Magyar, PhD.
ak. rok. 2021/2022 ZS

Opakovanie

- princípy OOP
 - abstrakcia
 - zapuzdrenie (enkapsulácia)
 - dedenie
 - polymorfizmus

Opakovanie

čo je stav objektu?

čo je správanie objektu?

čo je rozhranie objektu?

čo je kontrakt objektu?

aké sú modifikátory viditeľnosti?

ako sa správa podtrieda, ktorá dedí funkcionálnosť od nadtriedy?

Problémy so základným dedením

Čo ak nadtrieda je iba koncept, z ktorého nepotrebujeme vytvoriť inštanciu?

Čo ak nechceme zadať celú funkcionálnosť nadtriedy?

Čo ak v nadtriede nechceme implementovať žiadnu funkcionálnosť?

Abstraktná trieda

popis abstraktných konceptov

z abstraktnej triedy nemôžeme vytvoriť inštanciu

garantuje, že podtriedy prepíšu abstraktné metódy

v C# nemusí mať ani jednu abstraktnú metódu

Abstraktná metóda

nemá telo

iba deklarovaná

musí byť implementovaná v neabstraktných podtriedach

iba v abstraktných triedach

implicitne `virtual`

Kľúčové slovo `abstract`

v C# použiteľné pre triedy, metódy, indexery, udalosti (eventy) a properties

súvisiace kľúčové slovo - `sealed`

Používanie abstraktných tried

pri dedení, definícia hierarchie tried

definícia spoločnej funkcionality v rámci hierarchie

nesmie byť statická

iba priame dedenie

definícia typov parametrov

Konštruktor v abstraktnej triede

abstraktná trieda môže definovať niekoľko konštruktorov

konštruktor podtriedy by vždy mal zavolať konštruktor nadtriedy

konštruktory by mali byť `protected`

Hierarchia tried

umožnená cez dedenie

umožňuje jednotný pohľad na objekty rôznych tried

podpora polymorfizmu

zvyčajne hlavná trieda, ktorá je nadtriedou všetkých ostatných tried

v C# je to `System.Object`

Výhody hierarchie tried

polymorfické referencie

znovapoužitie kódu

umožňuje definíciu univerzálnych atribútov a správání ako
porovnávanie, vytvorenie kópií, hašovanie

System.Object

Equals (Object)

Finalize ()

GetHashCode ()

GetType ()

MemberwiseClone ()

ReferenceEquals (Object, Object)

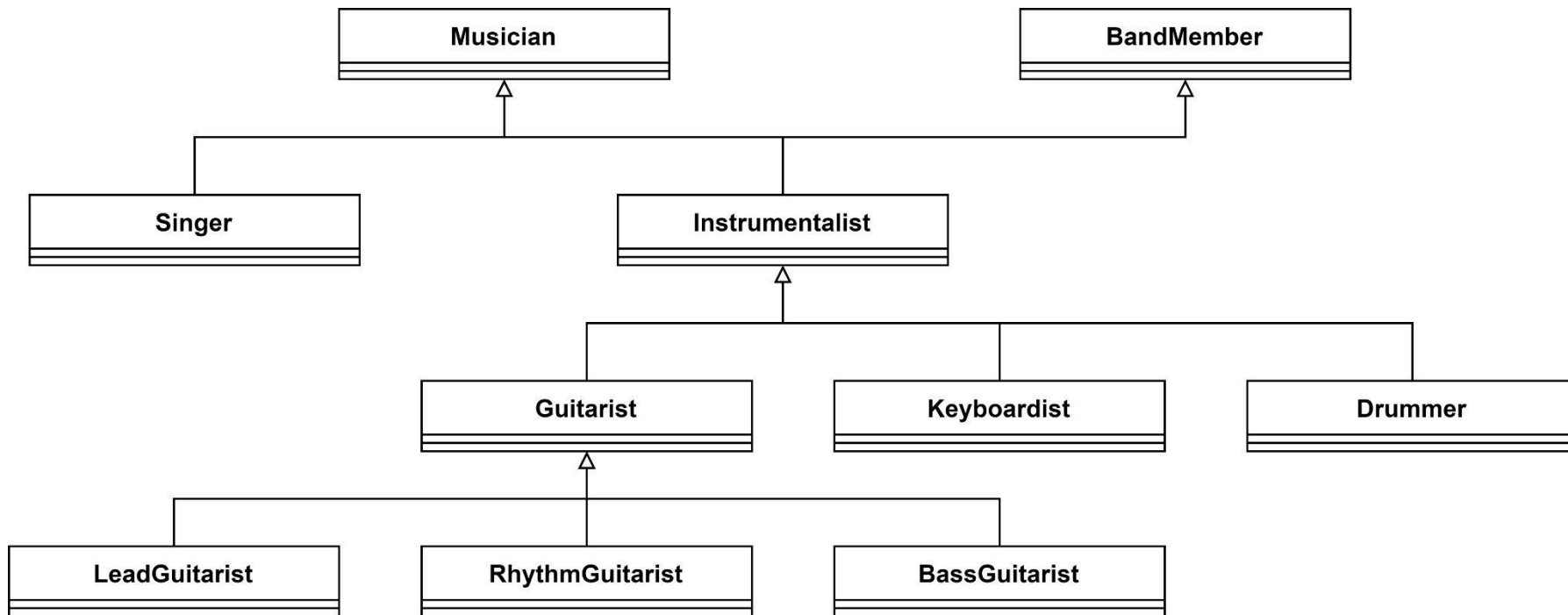
ToString ()

Vytvorenie hierarchie tried

cez dedenie (:)

každá trieda je automaticky podtriedou `System.Object`

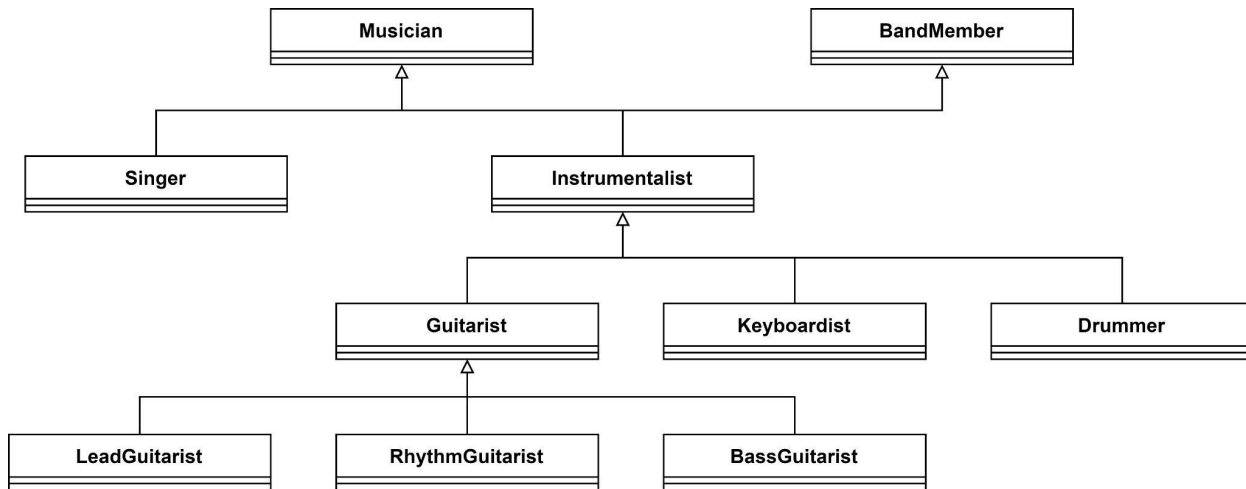
Hierarchia tried - ukážka



Viacnásobné dedenie

```
public class Musician {  
    ...  
    public bool PlaysInstrument() {  
        return false;  
    }  
    ...  
}
```

```
public class Instrumentalist {  
    ...  
    public bool PlaysInstrument() {  
        return true;  
    }  
    ...  
}
```



Vyriešenie problému viacnásobného dedenia

umožni viacnásobné dedenie - C++, Python, ...

neumožni viacnásobné dedenie - Java, C#, ...

Rozhranie

rozhranie (**interface**) je definícia kontraktu: každá trieda implementujúca rozhranie musí zadať metódy rozhrania podobné abstraktnej triede

rozhranie obsahuje iba abstraktné metódy a konštanty

Rozhrania v C#

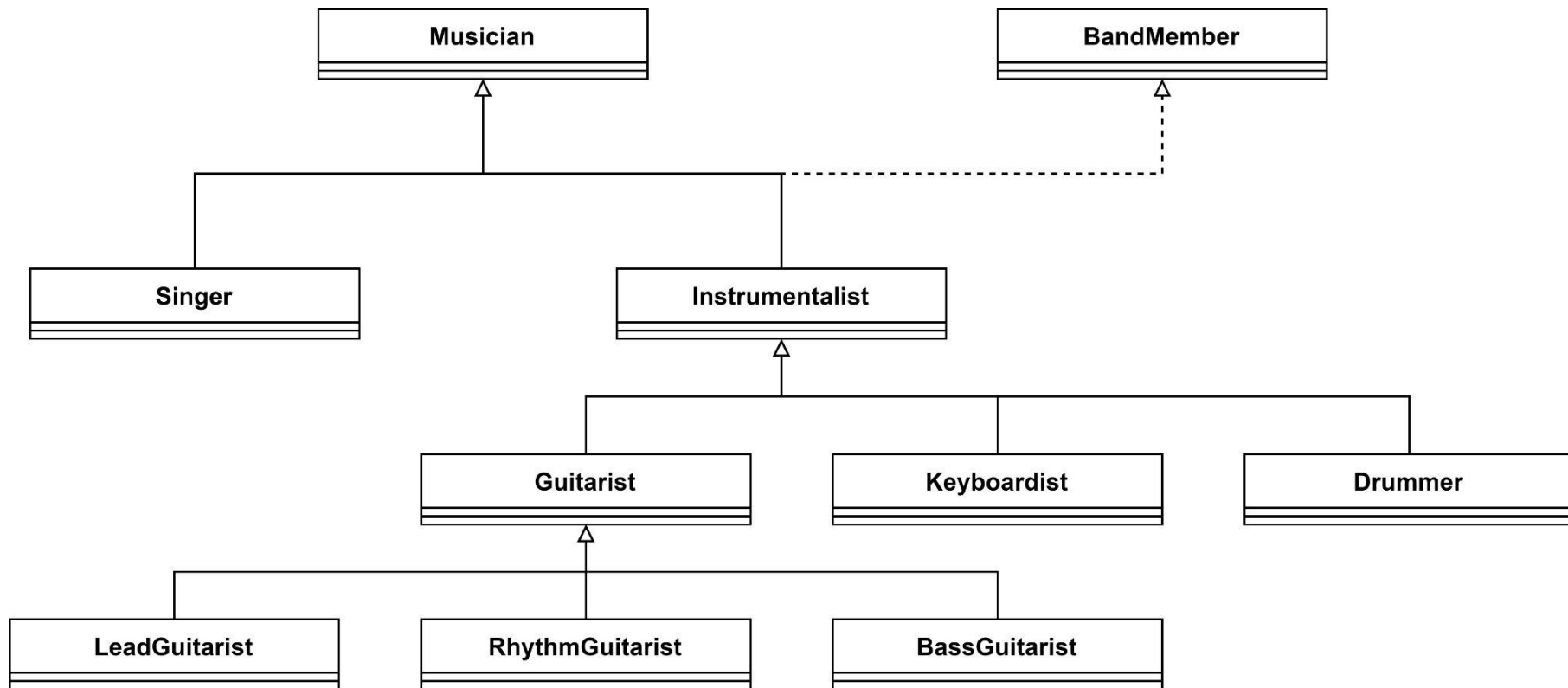
definícia cez kľúčové slovo interface

od C# 8.0 môže obsahovať defaultné implementácie (neodporúča sa)

môže definovať statické prvky pre jednu implementáciu spoločnej funkcionality

môže obsahovať metódy, atribúty, indexery, eventy, konštanty, operátory, statické konštruktory, vnorené typy, statické prvky, explicitné modifikátory viditeľnosti

Hierarchia tried s rozhraním



Aplikácia rozhraní

označenie ako s dedením

rozhranie je súbor všetkých metód, ktoré trieda musí implementovať

trieda môže implementovať viacero rozhraní

rozhranie môže dediť od iného rozhrania

z rozhrania nemôžeme vytvoriť inštanciu

Rozhranie vs typ

typ je rozhranie objektu - definuje vlastnosti a schopnosti

viaceré objekty môžu byť rovnakého typu a ten istý objekt môže mať niekoľko typov - polymorfizmus

objekt je prístupný cez svoje rozhranie

rozhranie je implementácia subtypingu v OOP

Rozhranie vs abstraktná trieda

C# umožňuje dedenie iba od jednej triedy

rozhranie nemôže obsahovať neabstraktné metódy

triedy implementujúce rozhranie musia zdefinovať všetky metódy

abstraktná trieda môže obsahovať abstraktné aj neabstraktné prvky

Rozhranie vs abstraktná trieda

pre rozhranie nemôžeme vytvoriť inštanciu

rozhranie je aj typom

rozhranie je spôsob podpory polymorfizmu

Rozhranie v štruktúre tried

jedna trieda môže implementovať niekoľko rozhraní

rozhranie nie je nadtriedou, pridáva funkcionality

dedenie je silný vzťah IS-A

implementácia je slabý vzťah IS-KIND-OF

Rozhrania best practice

zvyčajne malý počet metód

názov sa začína na I

názov je zvyčajne prídavné meno: IGroupable

alebo podstatné meno: IShape

otázky?