

Contents

Contents

1 Inheritance	1
1.1 Inheritance Properties	1
2 When to Use Inheritance?	3

1 Inheritance

Inheritance

- Most of classes in real life can be ordered in some type of hierarchy
 - Fridge, coffee-maker, microwave and others are *kitchen appliances*
 - Kitchen appliances, computers, audiotechnic and others are *electrical appliances*
 - ...
- For this type of hierarchy are typical ISA relations (ISA = is a)
- Hierarchy can be written by graph of classes

ISA Relation

- ISA relation:
 - All dogs are animals, all birds are animals, all reptiles are animals, ...
 - ISA relation is not *symetric* – Not all animals are dogs (or birds, or reptiles, ...); it is *antisymetric*
 - ISA relation is *transitive* – All mammals are animals, therefore all dogs are animals
 - ISA relation is *reflexive* – All dogs are dogs
 - Mathematically said - ISA relation is partial ordering

1.1 Inheritance Properties

Class Properties in Hierarchy

What relation is between classes "dog" and "animal"?

- Dog has all animals' attributes
 - ...and probably some more attributes

- Dog has all animals' methods
 - ... and probably some more methods
 - Some methods can differ in implementation (more later)

Terminology of OOP

- Relation between animal and dog is called *inheritance* (č. dědění)
 - Dog takes (inherits) all its' attributes and methods
- Dog is in relation to mammal called *descendant* (č. potomek)
- Mammal is in relation to dog called *ancestor* (č. předek)

Using inheritance in Pascal

```

type CAnimal=object
    private
        Weight:Real;
        ...
    public
        procedure feed;
        procedure sleep(howLong:Real);
        ...
end;
CDog=object(CMammal)
    private
        ...
    public
        procedure bark;
        ...
end;

```

Object Inheritance Limitations

- You can inherit only from one ancestor
 - There are (rare) languages with multiple inheritance – e.g. C++
- Neither attributes nor methods can be forgotten
 - Methods' implementation can be altered – through redefining or polymorphism (more later)
 - Some good object-oriented languages (Java, C++, not Pascal) can distinguish between methods (and functions) according to number and type of parameters or return type

2 When to Use Inheritance?

When to Use Inheritance?

- If there is hierarchical ISA relation between classes
 - One class is specialisation of other one

Right usage:

- Class *man* is ancestor for classes *employee* and *customer*
 - Because both employee and customer are mans...

Wrong usage:

- Class *point* is ancestor for classes *circle* or *vector*
 - *Neither* circle *nor* vector is point
 - Circle (and vector) can contain point – but there is other type of relation, not ISA