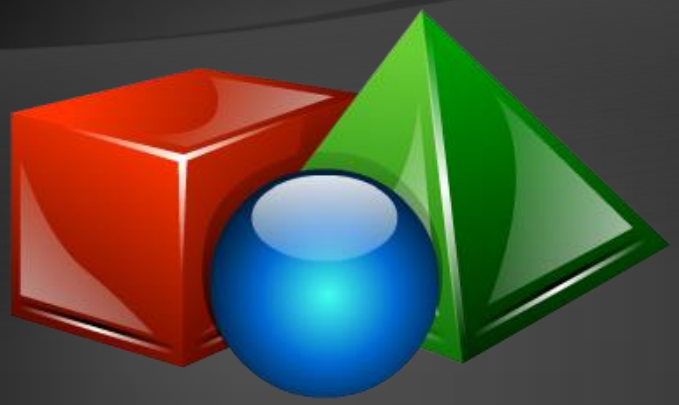
Telerik Academy



Using Objects

Objects, Properties, Primitive and Reference Types

Telerik Software Academy

Learning & Development Team

http://academy.telerik.com

Telerik Academy

Table of Contents

Telerik Academy

Object Types and Objects

Modeling Real-world Entities with Objects

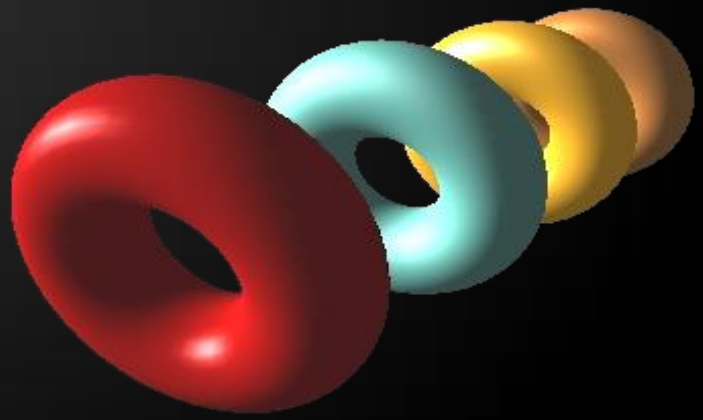
Telerik Academy

What are Objects?

- Software objects model real-world objects or abstract concepts
  - Examples:
    - bank, account, customer, dog, bicycle, queue
- Real-world objects have states and behaviors
  - Account' states:
    - holder, balance, type
  - Account' behaviors:
    - withdraw, deposit, suspend

What are Objects? (2)

- How do software objects implement real-world objects?
  - Use variables/data to implement states
  - Use methods/functions to implement behaviors
- An object is a software bundle of variables and related methods



Objects Represent

- checks
- people
- shopping list

Things from the real world

...

- numbers
- characters
- queues
- arrays

Things from the computer world

What is a Class?

- The formal definition of a object type:

> Object types act as templates from which an instance of an object is created at run time. Types define the properties of the object and the methods used to control the object's behavior.

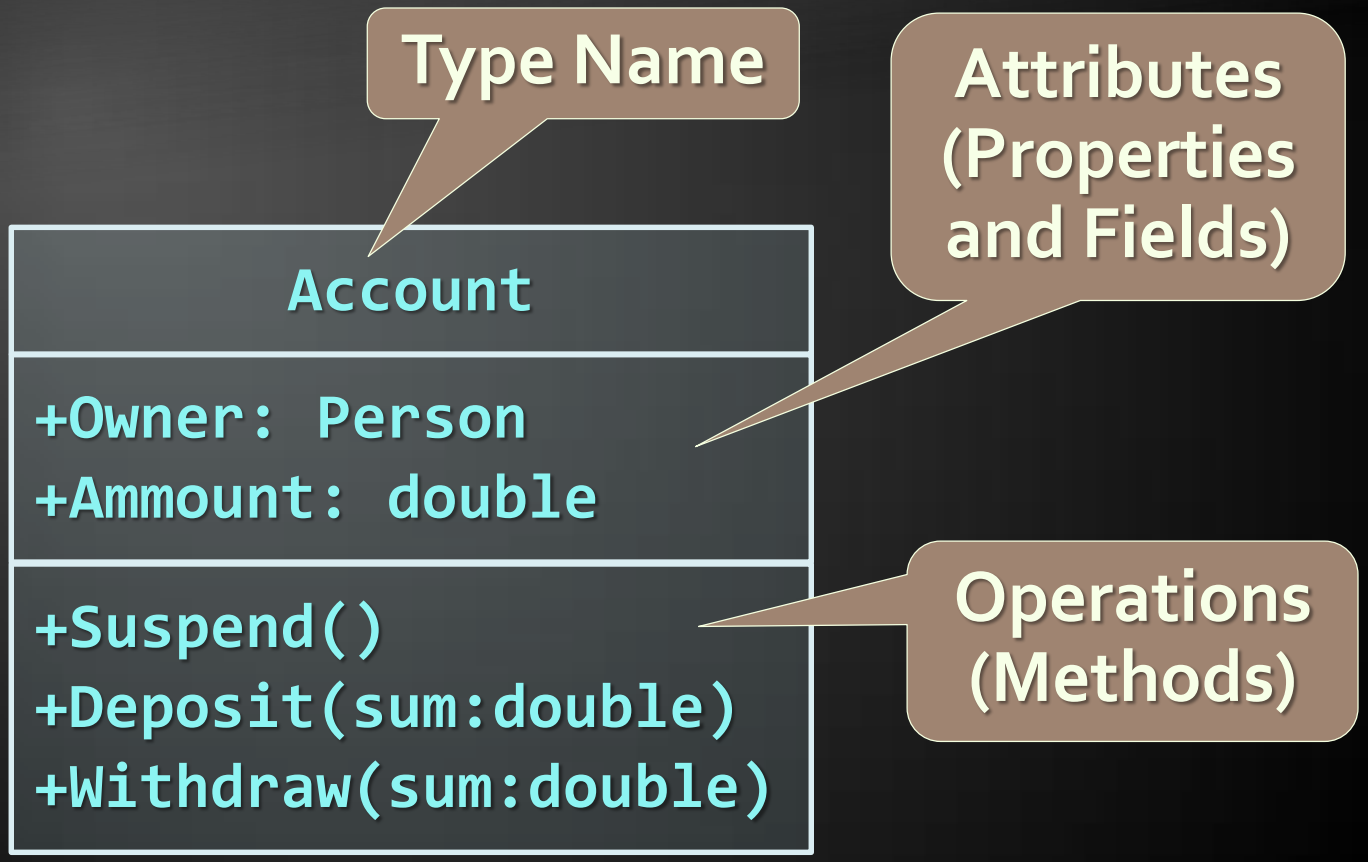
Definition by Google

Object Types

- Object Types provide the structure for objects
  - Define their prototype, act as template
- Object Types define:
  - Set of attributes
    - Represented by variables and properties
    - Hold their state
  - Set of actions (behavior)
    - Represented by methods
- A type defines the methods and types of data associated with an object

Telerik Academy

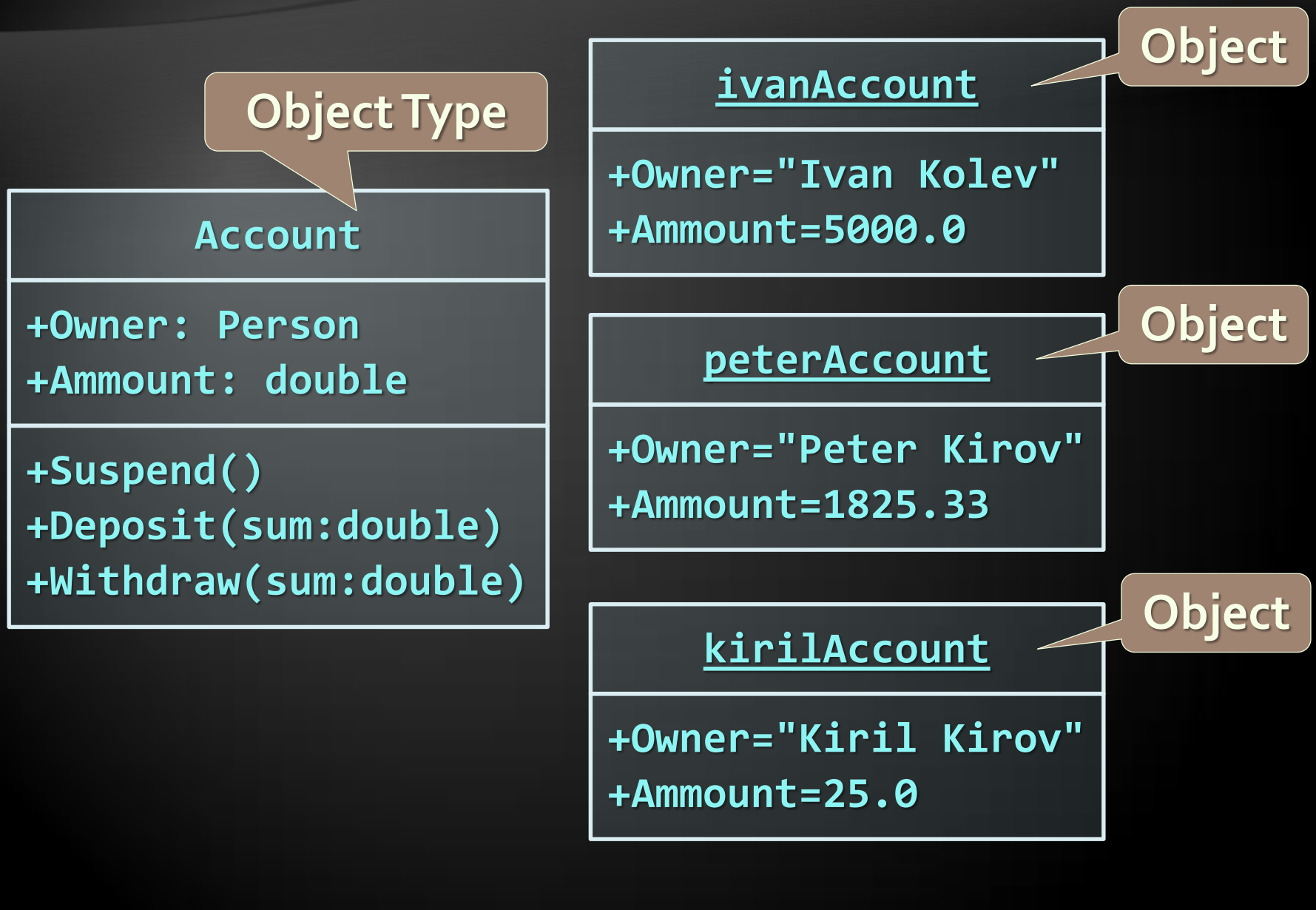
Object Types – Example



Objects

- An object is a concrete instance of a particular object type
- Creating an object from an object type is called instantiation
- Objects have state
  - Set of values associated to their attributes
- Example:
  - Type: Account
  - Objects: Ivan's account, Peter's account

Objects – Example



Telerik Academy

JavaScript Objects Overview

What are Objects?

Telerik Academy

Objects Overview

- JavaScript is designed on a simple object-based paradigm
  - An object is a collection of properties
- An object property is association between a name and a value
  - A value of property can be either a method (function) or a field (variable)
- Lots of predefined objects available in JS
  - Math, document, window, etc…
- Objects can be created by the developer

Telerik Academy

Object Properties

- Each object has properties
  - Properties are variables attached to the object
  - Properties of an object can be accessed with a dot-notation:

```
var arrStr = arr.join(', '); // property join of Array
var length = arr.length;  // property length of Array
var words = text.split(' ');
```

Telerik Academy

Objects and Properties

Live Demo

Telerik Academy

Object and Primitive Types

The Types in JavaScript

Telerik Academy

Reference and Primitive Types

- JavaScript is a typeless language
  - Variables don't have type, but their values do
- JavaScript has six different types:
  - `Number`, `String`, `Boolean`, `Null`, `Undefined` and `Object`
- `Object` is the only Object type
  - It is copied by reference
- `Number`, `String`, `Boolean`, `Null`, `Undefined` are primitive types
  - Copied by value

Telerik Academy

Reference and Primitive Types (2)

- The primitive types are `Boolean`, `Number`, `String`, `Undefined` and `Null`
  - All the other types are actually of type object
    - Including arrays, dates, custom types, etc…

```
console.log(typeof new Object() === typeof new Array()); // true
console.log(typeof new Object() === typeof new Date()); // true
console.log(typeof new Array() === typeof new Date()); // true
```

- All types derive from object
  - Their type is object

Telerik Academy

Primitive and Reference Types

Live Demo

Primitive Types

- Primitive types are passed by value
  - When passed as argument
    - New memory is allocated (in the stack)
    - The value is copied in the new memory
    - The value in the new memory is passed
- Primitive types are initialized with type literals

```
var number = 5;
var text = 'Hello there!';
```

- Primitive types have a object type wrapper

```
var number = 5; // Holds a primitive value of 5
var numberObj = new Number(5); // Holds a object value of 5
```

Telerik Academy

Primitive Types – Example

- Assign string values to two variables
  - Create an object using their value
  - Change the value of the variables
  - Each object has its own value

```
var fname = 'Pesho';
var lname = 'Ivanov';

var person = { firstName: fname, lastName: lname };

lname = 'Petrov';
console.log(person.lastName) // logged 'Ivanov'
```

Telerik Academy

Primitive Types

Live Demo

Reference Type

- Object is the only object type
  - When passed to a function the value is not copied, but instead a reference of it is passed

```
var marks= [
  { subject : 'JavaScript', score : 4.50 },
  { subject : 'OOP', score : 5.00 },
  { subject : 'HTML5', score : 6.00 },
  { subject : 'Photoshop', score : 4.00 }];

var student = { name: 'Doncho Minkov', marks: marks };
marks[2].score = 5.50;

console.log(student.marks); // logs 5.50 for HTML5 score
```

Telerik Academy

Object Types

Live Demo

Telerik Academy

JSON Objects

Creating Simple objects

Telerik Academy

JSON Objects

- JSON stands for JavaScript Object Notation
  - A data format used in JavaScript

```
var person = {
  firstName: 'Doncho',
  lastName: 'Minkov',
  toString: function personToString() {
    return this.firstName + ' ' + this.lastName;
  }
}
```

  - Then the object properties can be used:

```
console.log(person.toString()); // writes 'Doncho Minkov'
```

Telerik Academy

JSON Objects

Live Demo

Telerik Academy

Building a JSON Object

- JSON is great, but repeating code is not, right?
  - Lets make two persons:

```
var minkov = {fname: 'Doncho', lname: 'Minkov',
  toString: function(){ return this.fname + ' ' + this.lname;}
}
var georgiev = { fname: 'Georgi', lname: 'Georgiev',
  toString: function(){ return this.fname + ' ' + this.lname;}
}
```

  - Lots of repeating code
    - Can't we use a constructor or function to create an object?

Telerik Academy

JSON Building Function

- A function for building JSON objects
  - Just pass first and last name and get a object
    - Something like a constructor

```
function buildPerson(fname, lname) {
  return {
    fname: fname,
    lname: lname,
    toString: function (){return this.fname + ' ' + this.lname;}
  }
}
var minkov = buildPerson('Doncho', 'Minkov');
var georgiev = buildPerson('Georgi', 'Georgiev');
```

- Much cooler, right?

Telerik Academy

JSON Building Function

Live Demo

Telerik Academy

JavaScript Object Properties

Telerik Academy

JS Object Properties

- JavaScript objects are just a set of key/value pairs
  - Each value can be accessed by its key
- All objects in JavaScript are parsed to JSON
  - Properties in JSON are accessed using the dot-notation (obj.property)
  - Yet properties can be used with brackets
    - Like an array

```
document.write === document['write'] // results in true
```

Telerik Academy

JavaScript Object Properties

Live Demo

Telerik Academy

Associative Arrays

- Objects can be used as associative arrays
  - The key (index) is string instead of number
    - Also called `dictionaries` or `maps`
- Associative arrays don't have array properties
  - `length`, `indexOf`, etc…

```
function countWords(words) {
  var wordsCount = {};
  for (var i in words) {
    var word = words[i].toLowerCase();
    if (wordsCount[word]) wordsCount[word]++;
    else wordsCount[word] = 1;
  }
  return wordsCount
}
```

Telerik Academy

Associative Arrays

Live Demo

Telerik Academy

Using Objects

Questions?

http://academy.telerik.com

Homework

1. Write functions for working with shapes in standard Planar coordinate system
   - Points are represented by coordinates P(X, Y)
   - Lines are represented by two points, marking their beginning and ending
     - L($P_1(X_1,Y_1)$, $P_2(X_2,Y_2)$)
   - Calculate the distance between two points
   - Check if three segment lines can form a triangle

Telerik Academy

Homework (2)

2. Write a function that removes all elements with a given value

```
var arr = [1,2,1,4,1,3,4,1,111,3,2,1,'1'];
arr.remove(1); //arr = [2,4,3,4,111,3,2,'1'];
```

   - Attach it to the array type
   - Read about prototype and how to attach methods
3. Write a function that makes a deep copy of an object
   - The function should work for both primitive and reference types
4. Write a function that checks if a given object contains a given property

```
var obj  = …;
var hasProp = hasProperty(obj, 'length');
```

Homework (3)

5. Write a function that finds the youngest person in a given array of persons and prints his/hers full name
   - Each person have properties firstname, lastname and age, as shown:

```
var persons = [
  { firstname : 'Gosho', lastname: 'Petrov', age: 32 },
  { firstname : 'Bay', lastname: 'Ivan', age: 81},… ];
```

6. Write a function that groups an array of persons by age, first or last name. The function must return an associative array, with keys - the groups, and values - arrays with persons in this groups
   - Use function overloading (i.e. just one function)

```
var persons = {…};
var groupedByFname = group(persons, 'firstname');
var groupedByAge= group(persons, 'age');
```