

C# Language & .NET Platform

8th Lecture:

C# Type System

<http://d3s.mff.cuni.cz/~jezek>

Department of
Distributed and
Dependable
Systems



Pavel Ježek

pavel.jezek@d3s.mff.cuni.cz



CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

Which variant is faster?

```
partial class Program {
    private static void f() {
        T.m(1);
    }

    private static void g() {
        for (int i = 0; i < 100000000; i++) {
            T.m(1);
        }
    }

    private static void doCalls1() {
        f();
        g();
    }

    private static void doCalls2() {
        g();
        f();
    }
}
```

```
partial class Program {
    static void Main(string[] args) {
        doCalls1();
        for (int j = 0; j < 10; j++) {
            Stopwatch sw = System.Diagnostics.Stopwatch.StartNew();
            doCalls1();
            sw.Stop();
            Console.WriteLine(sw.Elapsed.ToString());
        }
    }
}
```

A

```
partial class Program {
    static void Main(string[] args) {
        doCalls2();
        for (int j = 0; j < 10; j++) {
            Stopwatch sw = System.Diagnostics.Stopwatch.StartNew();
            doCalls2();
            sw.Stop();
            Console.WriteLine(sw.Elapsed.ToString());
        }
    }
}
```

B

Speed of `T.m(int a)` does not depend on its argument value.

Option	Result
A	None – A and B are equivalent.
B	<i>A is always faster than B.</i>
C	<i>B is always faster than A.</i>
D	<i>A may be faster than B (depending on implementation of T).</i>
E	<i>B may be faster than A (depending on implementation of T).</i>

What is the output the following program?

```
class A {
    public string className = "A";
}
class B : A {
    private string className = "B";
}
class Program {
    static void Main(string[] args) {
        Console.WriteLine(new B().className);
    }
}
```

Option	Result
A	<i>It will not compile – error in class B.</i>
B	<i>It will not compile – error in class Program.</i>
C	A
D	B
E	<i>It will generate a runtime error.</i>

What is the output the following program?

```
class A {
    public string className = "A";
}
class B : A {
    private new string className = "B";
}
class Program {
    static void Main(string[] args) {
        Console.WriteLine(new B().className);
    }
}
```

Option	Result
A	It will not compile - error in class B.
B	It will not compile - error in class Program.
C	A
D	B
E	It will generate a runtime error.

a compiler warning: use new keyword

What is the output the following program?

```
class A {
    public string className = "A";
}
class B : A {
    private string className = "B";
}
class Program {
    static void Main(string[] args) {
        Console.WriteLine(new B().className);
    }
}
```

Option	Result
A	<i>It will not compile – error in class B.</i>
B	<i>It will not compile – error in class Program.</i>
C	<i>A & a compiler warning: use new keyword.</i>
D	B
E	<i>It will generate a runtime error.</i>

What is the output the following program?

```
class A {
    public string className = "A";
}
class B : A {
    private string className = "B";
    public void f() {
        Console.WriteLine(className);
    }
}
class Program {
    static void Main(string[] args) {
        Console.Write(new B().className);
        new B().f();
    }
}
```

Option	Result
A	A A
B	A B
C	B A
D	B B

What is the output the following program?

```
class A {
    public string className = "A";
}
class B : A {
    private string className = "B";
    public void f() {
        Console.WriteLine(className);
    }
}
class Program {
    static void Main(string[] args) {
        Console.Write(new B().className);
        new B().f();
    }
}
```

Option	Result
A	A A
B	A B & a compiler warning: <i>use new keyword.</i>
C	B A
D	B B

What is the output the following program?

```
class Program {
    Program(object o) {
        Console.WriteLine("Program(object o)");
    }
    Program(int[] a) {
        Console.WriteLine("Program(int[] a)");
    }
    static void Main(string[] args) {
        new Program(null);
    }
}
```

Option	Result
A	<i>It will not compile.</i>
B	Program(object o)
C	Program(int[] a)
D	<i>It will generate a runtime error.</i>

What is the output the following program?

```
class A {}
class B : A{}
class Program {
    Program(A a) {
        Console.WriteLine("Program(A a)");
    }
    Program(B b) {
        Console.WriteLine("Program(B b)");
    }
    static void Main(string[] args) {
        new Program(new B());
    }
}
```

Option	Result
A	<i>It will not compile.</i>
B	Program(A a)
C	Program(B b)
D	<i>It will generate a runtime error.</i>

What is the output the following program?

```
class A {}
class B : A{}
class Program {
    Program(A a) {
        Console.WriteLine("Program(A a)");
    }
    Program(B b) {
        Console.WriteLine("Program(B b)");
    }
    static void Main(string[] args) {
        new Program(new B());
    }
}
```

Option	Result
A	<i>It will not compile.</i>
B	Program(A a)
C	Program(B b)
D	<i>It will generate a runtime error.</i>

What is the output the following program?

```
class Program {
    Program(object o) {
        Console.WriteLine("Program(object o)");
    }
    Program(int[] a) {
        Console.WriteLine("Program(int[] a)");
    }
    static void Main(string[] args) {
        new Program(null);
    }
}
```

Option	Result
A	<i>It will not compile.</i>
B	Program(object o)
C	Program(int[] a)
D	<i>It will generate a runtime error.</i>

What is the output the following program?

```
class Program {
    Program(object o) {
        Console.WriteLine("Program(object o)");
    }
    Program(int[] a) {
        Console.WriteLine("Program(int[] a)");
    }
    static void Main(string[] args) {
        new Program(null);
    }
}
```

Option	Result
A	<i>It will not compile.</i>
B	Program(object o)
C	Program(int[] a)
D	<i>It will generate a runtime error.</i>

What is the output the following program?

```
class A {}
class B : A{}
class Program {
    Program(object o) {
        Console.WriteLine("Program(object o)");
    }
    Program(int[] a) {
        Console.WriteLine("Program(int[] a)");
    }
    Program(A a) {
        Console.WriteLine("Program(A a)");
    }
    Program(B b) {
        Console.WriteLine("Program(B b)");
    }
    static void Main(string[] args) {
        new Program(null);
    }
}
```

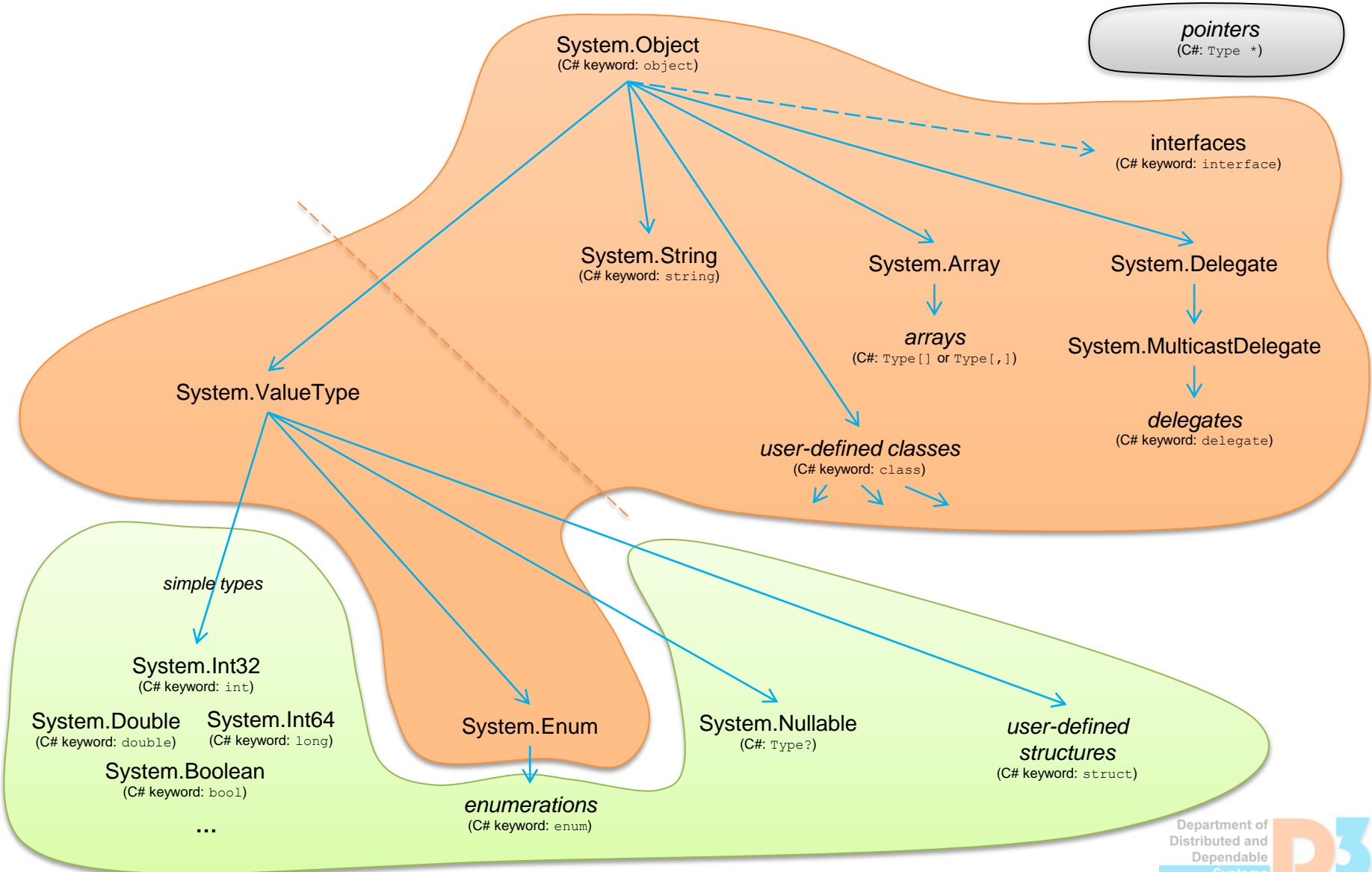
Option	Result
A	<i>It will not compile.</i>
B	Program(object o)
C	Program(int[] a)
D	Program(A a)
E	Program(B b)
F	<i>It will generate a runtime error.</i>

What is the output the following program?

```
class A {}
class B : A{}
class Program {
    Program(object o) {
        Console.WriteLine("Program(object o)");
    }
    Program(int[] a) {
        Console.WriteLine("Program(int[] a)");
    }
    Program(A a) {
        Console.WriteLine("Program(A a)");
    }
    Program(B b) {
        Console.WriteLine("Program(B b)");
    }
    static void Main(string[] args) {
        new Program(null);
    }
}
```

Option	Result
A	<i>It will not compile: The call is ambiguous between the following methods or properties: 'Program.Program(B)' and 'Program.Program(int[])'</i>
B	Program(object o)
C	Program(int[] a)
D	Program(A a)
E	Program(B b)
F	<i>It will generate a runtime error.</i>

CLI Type Inheritance



Constructors and Inheritance



Implicit call of the base class constructor

Explicit call

```
class A {  
    public A() {...}  
}
```

```
class B : A {  
    ...  
}
```

```
B b = new B();
```

OK

- A()
- Default-constr. B()

```
class A {  
    ...  
}
```

```
class B : A {  
    public B(int x) {...}  
}
```

```
B b = new B(3);
```

OK

- Default-constr. A()
- B(int x)

```
class A {  
    public A() {...}  
}
```

```
class B : A {  
    public B(int x) {...}  
}
```

```
B b = new B(3);
```

OK

- A()
- B(int x)

```
class A {  
    public A(int x) {...}  
}
```

```
class B : A {  
    public B(int x) {...}  
}
```

```
B b = new B(3);
```

Error!

- no explicit call of the A() constructor
- default constr. A() does not exist

```
class A {  
    public A(int x) {...}  
}
```

```
class B : A {  
    public B(int x)  
        : base(x) {...}  
}
```

```
B b = new B(3);
```

OK

- A(int x)
- B(int x)

Which variant is faster?

```
partial class Program {
    private static void f() {
        T.m(1);
    }

    private static void g() {
        for (int i = 0; i < 100000000; i++) {
            T.m(1);
        }
    }

    private static void doCalls1() {
        f();
        g();
    }

    private static void doCalls2() {
        g();
        f();
    }
}
```

```
partial class Program {
    static void Main(string[] args) {
        doCalls1();
        for (int j = 0; j < 10; j++) {
            Stopwatch sw = System.Diagnostics.Stopwatch.StartNew();
            doCalls1();
            sw.Stop();
            Console.WriteLine(sw.Elapsed.ToString());
        }
    }
}
```

A

```
partial class Program {
    static void Main(string[] args) {
        doCalls2();
        for (int j = 0; j < 10; j++) {
            Stopwatch sw = System.Diagnostics.Stopwatch.StartNew();
            doCalls2();
            sw.Stop();
            Console.WriteLine(sw.Elapsed.ToString());
        }
    }
}
```

B

Speed of `T.m(int a)` does not depend on its argument value.

Option	Result
A	None – A and B are equivalent.
B	<i>A is always faster than B.</i>
C	<i>B is always faster than A.</i>
D	<i>A may be faster than B (depending on implementation of T).</i>
E	<i>B may be faster than A (depending on implementation of T).</i>

Which variant is faster?

```
partial class Program {
    private static void f() {
        possible CLR.CallStaticConstructorOnce
        T.m(1);
    }

    private static void g() {
        for (int i = 0; i < 100000000; i++) {
            possible CLR.CallStaticConstructorOnce
            T.m(1);
        }
    }

    private static void doCalls1() {
        f();
        g();
    }

    private static void doCalls2() {
        g();
        f();
    }
}
```

```
partial class Program {
    static void Main(string[] args) {
        doCalls1();
        for (int j = 0; j < 10; j++) {
            Stopwatch sw = System.Diagnostics.Stopwatch.StartNew();
            doCalls1();
            sw.Stop();
            Console.WriteLine(sw.Elapsed.ToString());
        }
    }
}
```

A

```
partial class Program {
    static void Main(string[] args) {
        doCalls2();
        for (int j = 0; j < 10; j++) {
            Stopwatch sw = System.Diagnostics.Stopwatch.StartNew();
            doCalls2();
            sw.Stop();
            Console.WriteLine(sw.Elapsed.ToString());
        }
    }
}
```

B

Option	Result
A	None – A and B are equivalent.
B	<i>A is always faster than B.</i>
C	<i>B is always faster than A.</i>
D	<i>A may be faster than B (if T has a static constructor).</i>
E	<i>B may be faster than A (depending on implementation of T).</i>

Which variant is faster?

A

```
partial class Program {
    private static void f() {
        call CLR.CallStaticConstructorOnce
        T.m(1);
    }

    private static void g() {
        for (int i = 0; i < 100000000; i++) {
            T.m(1);
        }
    }

    private static void doCalls1() {
        f();
        g();
    }

    private static void doCalls2() {
        g();
        f();
    }
}
```

A

```
partial class Program {
    static void Main(string[] args) {
        doCalls1();
        for (int j = 0; j < 10; j++) {
            Stopwatch sw = System.Diagnostics.Stopwatch.StartNew();
            doCalls1();
            sw.Stop();
            Console.WriteLine(sw.Elapsed.ToString());
        }
    }
}
```

B

```
partial class Program {
    static void Main(string[] args) {
        doCalls2();
        for (int j = 0; j < 10; j++) {
            Stopwatch sw = System.Diagnostics.Stopwatch.StartNew();
            doCalls2();
            sw.Stop();
            Console.WriteLine(sw.Elapsed.ToString());
        }
    }
}
```

Option	Result
A	None – A and B are equivalent.
B	<i>A is always faster than B.</i>
C	<i>B is always faster than A.</i>
D	<i>A may be faster than B (if T has a static constructor).</i>
E	<i>B may be faster than A (depending on implementation of T).</i>