



Created for <http://bugs.freepascal.org/view.php?id=25607>

by Maciej Izak (aka. hnb) hnb.code@gmail.com

Version for „no override” in class TB;

Legend	
	Should be fixed
	Expected behavior

## Tests for constructors

Tests for constructors (based on constructors/A0X.dpr)

ID	TA = class		Delphi tested version (same results) 2007, XE2 (32 and 64 bit), XE4 (32 and 64 bit)			FPC 2.7.1		
	Overloaded Constructors		Compiles	TObject.Create is hidden	TA.Create(); will call	Compiles	TObject.Create is hidden	TA.Create(); will call
1	Create(A: Integer = 0)	Create(A: string = "")	NO			NO		
2	Create(A: Integer = 0)	Create(A: string)	YES	YES	TA.Create(A: Integer = 0)	YES	YES	TA.Create(A: Integer = 0)
3	Create(A: Integer)	Create(A: string = "")	YES	YES	TA.Create(A: string = "")	YES	YES	TA.Create(A: string = "")
4	Create(A: Integer)	Create(A: string)	YES	NO	TObject.Create	YES	NO	TObject.Create

Tests for constructors (based on constructors/BXX.dpr)

ID	TB = class(TA)		Inherited version of class TA	Delphi tested version (same results) 2007, XE2 (32 and 64 bit), XE4 (32 and 64 bit)			FPC 2.7.1		
	Overloaded Constructors (no override)			Compiles	TObject.Create is hidden	TB.Create(); will call	Compiles	TObject.Create is hidden	TB.Create(); will call
1	Create(A: Integer = 0)	Create(A: string = "")	1	NO			NO		
2	Create(A: Integer = 0)	Create(A: string = "")	2	NO			NO		
3	Create(A: Integer = 0)	Create(A: string = "")	3	NO			NO		
4	Create(A: Integer = 0)	Create(A: string = "")	4	NO			NO		
5	Create(A: Integer = 0)	Create(A: string)	1	YES	YES	TB.Create(A: Integer = 0)	YES	YES	TB.Create(A: Integer = 0)
6	Create(A: Integer = 0)	Create(A: string)	2	YES	YES	TB.Create(A: Integer = 0)	YES	YES	TB.Create(A: Integer = 0)
7	Create(A: Integer = 0)	Create(A: string)	3	YES	YES	TB.Create(A: Integer = 0)	YES	YES	TB.Create(A: Integer = 0)
8	Create(A: Integer = 0)	Create(A: string)	4	YES	YES	TB.Create(A: Integer = 0)	YES	YES	TB.Create(A: Integer = 0)
9	Create(A: Integer)	Create(A: string = "")	1	YES	YES	TB.Create(A: string = "")	YES	YES	TB.Create(A: string = "")
10	Create(A: Integer)	Create(A: string = "")	2	YES	YES	TB.Create(A: string = "")	YES	YES	TB.Create(A: string = "")
11	Create(A: Integer)	Create(A: string = "")	3	YES	YES	TB.Create(A: string = "")	YES	YES	TB.Create(A: string = "")
12	Create(A: Integer)	Create(A: string = "")	4	YES	YES	TB.Create(A: string = "")	YES	YES	TB.Create(A: string = "")
13	Create(A: Integer)	Create(A: string)	1	NO			NO		
14	Create(A: Integer)	Create(A: string)	2	YES	YES	TA.Create(A: Integer = 0)	NO		
15	Create(A: Integer)	Create(A: string)	3	YES	YES	TA.Create(A: string = "")	NO		
16	Create(A: Integer)	Create(A: string)	4	YES	NO	TObject.Create	YES	NO	TObject.Create
17	Create(A: Integer = 0)		1	YES	YES	TB.Create(A: Integer = 0)	YES	YES	TB.Create(A: Integer = 0)
18	Create(A: Integer = 0)		2	YES	YES	TB.Create(A: Integer = 0)	YES	YES	TB.Create(A: Integer = 0)
19	Create(A: Integer = 0)		3	YES	YES	TB.Create(A: Integer = 0)	YES	YES	TB.Create(A: Integer = 0)
20	Create(A: Integer = 0)		4	YES	YES	TB.Create(A: Integer = 0)	YES	YES	TB.Create(A: Integer = 0)
21	Create(A: Integer)		1	NO			NO		
22	Create(A: Integer)		2	YES	YES	TA.Create(A: Integer = 0)	NO		
23	Create(A: Integer)		3	YES	YES	TA.Create(A: string = "")	NO		
24	Create(A: Integer)		4	YES	NO	TObject.Create	YES	NO	TObject.Create
25		Create(A: string = "")	1	YES	YES	TB.Create(A: string = "")	YES	YES	TB.Create(A: string = "")
26		Create(A: string = "")	2	YES	YES	TB.Create(A: string = "")	YES	YES	TB.Create(A: string = "")
27		Create(A: string = "")	3	YES	YES	TB.Create(A: string = "")	YES	YES	TB.Create(A: string = "")
28		Create(A: string = "")	4	YES	YES	TB.Create(A: string = "")	YES	YES	TB.Create(A: string = "")
29		Create(A: string)	1	NO			NO		
30		Create(A: string)	2	YES	YES	TA.Create(A: Integer = 0)	NO		
31		Create(A: string)	3	YES	YES	TA.Create(A: string = "")	NO		
32		Create(A: string)	4	YES	NO	TObject.Create	YES	NO	TObject.Create

## Tests for methods

Tests for methods (based on methods/A0Xdpr)

ID	TA = class(T0)		Delphi tested version (same results) 2007, XE2 (32 and 64 bit), XE4 (32 and 64 bit)			FPC 2.7.1		
	Overloaded Methods		Compiles	T0.Foo is hidden	A.Foo(); will call	Compiles	T0.Foo is hidden	A.Foo(); will call
1	Foo(A: Integer = 0)	Foo(A: string = "")	NO			NO		
2	Foo(A: Integer = 0)	Foo(A: string)	YES	YES	TA.Foo(A: Integer = 0)	YES	YES	TA.Foo(A: Integer = 0)
3	Foo(A: Integer)	Foo(A: string = "")	YES	YES	TA.Foo(A: string = "")	YES	YES	TA.Foo(A: string = "")
4	Foo(A: Integer)	Foo(A: string)	YES	NO	T0.Foo	YES	NO	T0.Foo

Tests for methods(based on methods/BXXdpr)

ID	TB = class(TA)		Inherited version of class TA	Delphi tested version (same results) 2007, XE2 (32 and 64 bit), XE4 (32 and 64 bit)			FPC 2.7.1		
	Overloaded Methods (no override)			Compiles	T0.Foo is hidden	B.Foo(); will call	Compiles	T0.Foo is hidden	B.Foo(); will call
1	Foo(A: Integer = 0)	Foo(A: string = "")	1	NO			NO		
2	Foo(A: Integer = 0)	Foo(A: string = "")	2	NO			NO		
3	Foo(A: Integer = 0)	Foo(A: string = "")	3	NO			NO		
4	Foo(A: Integer = 0)	Foo(A: string = "")	4	NO			NO		
5	Foo(A: Integer = 0)	Foo(A: string)	1	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
6	Foo(A: Integer = 0)	Foo(A: string)	2	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
7	Foo(A: Integer = 0)	Foo(A: string)	3	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
8	Foo(A: Integer = 0)	Foo(A: string)	4	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
9	Foo(A: Integer)	Foo(A: string = "")	1	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
10	Foo(A: Integer)	Foo(A: string = "")	2	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
11	Foo(A: Integer)	Foo(A: string = "")	3	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
12	Foo(A: Integer)	Foo(A: string = "")	4	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
13	Foo(A: Integer)	Foo(A: string)	1	NO			NO		
14	Foo(A: Integer)	Foo(A: string)	2	YES	YES	TA.Foo(A: Integer = 0)	NO		
15	Foo(A: Integer)	Foo(A: string)	3	YES	YES	TA.Foo(A: string = "")	NO		
16	Foo(A: Integer)	Foo(A: string)	4	YES	NO	T0.Foo	YES	NO	T0.Foo
17	Foo(A: Integer = 0)		1	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
18	Foo(A: Integer = 0)		2	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
19	Foo(A: Integer = 0)		3	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
20	Foo(A: Integer = 0)		4	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
21	Foo(A: Integer)		1	NO			NO		
22	Foo(A: Integer)		2	YES	YES	TA.Foo(A: Integer = 0)	NO		
23	Foo(A: Integer)		3	YES	YES	TA.Foo(A: string = "")	NO		
24	Foo(A: Integer)		4	YES	NO	T0.Foo	YES	NO	T0.Foo
25		Foo(A: string = "")	1	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
26		Foo(A: string = "")	2	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
27		Foo(A: string = "")	3	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
28		Foo(A: string = "")	4	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
29		Foo(A: string)	1	NO			NO		
30		Foo(A: string)	2	YES	YES	TA.Foo(A: Integer = 0)	NO		
31		Foo(A: string)	3	YES	YES	TA.Foo(A: string = "")	NO		
32		Foo(A: string)	4	YES	NO	T0.Foo	YES	NO	T0.Foo

## Tests for class methods

Tests for class methods (based on classmethods/A0X.dpr)

ID	TA = class(T0)		Delphi tested version (same results) 2007, XE2 (32 and 64 bit), XE4 (32 and 64 bit)			FPC 2.7.1		
	Overloaded Class Methods		Compiles	T0.Foo is hidden	TA.Foo(); will call	Compiles	T0.Foo is hidden	TA.Foo(); will call
1	Foo(A: Integer = 0)	Foo(A: string = "")	NO			NO		
2	Foo(A: Integer = 0)	Foo(A: string)	YES	YES	TA.Foo(A: Integer = 0)	YES	YES	TA.Foo(A: Integer = 0)
3	Foo(A: Integer)	Foo(A: string = "")	YES	YES	TA.Foo(A: string = "")	YES	YES	TA.Foo(A: string = "")
4	Foo(A: Integer)	Foo(A: string)	YES	NO	T0.Foo	YES	NO	T0.Foo

Tests for class methods(based on classmethods/BXX.dpr)

ID	TB = class(TA)		Inherited version of class TA	Delphi tested version (same results) 2007, XE2 (32 and 64 bit), XE4 (32 and 64 bit)			FPC 2.7.1		
	Overloaded Class Methods (no override)			Compiles	T0.Foo is hidden	TB.Foo(); will call	Compiles	T0.Foo is hidden	TB.Foo(); will call
1	Foo(A: Integer = 0)	Foo(A: string = "")	1	NO			NO		
2	Foo(A: Integer = 0)	Foo(A: string = "")	2	NO			NO		
3	Foo(A: Integer = 0)	Foo(A: string = "")	3	NO			NO		
4	Foo(A: Integer = 0)	Foo(A: string = "")	4	NO			NO		
5	Foo(A: Integer = 0)	Foo(A: string)	1	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
6	Foo(A: Integer = 0)	Foo(A: string)	2	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
7	Foo(A: Integer = 0)	Foo(A: string)	3	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
8	Foo(A: Integer = 0)	Foo(A: string)	4	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
9	Foo(A: Integer)	Foo(A: string = "")	1	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
10	Foo(A: Integer)	Foo(A: string = "")	2	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
11	Foo(A: Integer)	Foo(A: string = "")	3	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
12	Foo(A: Integer)	Foo(A: string = "")	4	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
13	Foo(A: Integer)	Foo(A: string)	1	NO			NO		
14	Foo(A: Integer)	Foo(A: string)	2	YES	YES	TA.Foo(A: Integer = 0)	NO		
15	Foo(A: Integer)	Foo(A: string)	3	YES	YES	TA.Foo(A: string = "")	NO		
16	Foo(A: Integer)	Foo(A: string)	4	YES	NO	T0.Foo	YES	NO	T0.Foo
17	Foo(A: Integer = 0)		1	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
18	Foo(A: Integer = 0)		2	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
19	Foo(A: Integer = 0)		3	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
20	Foo(A: Integer = 0)		4	YES	YES	TB.Foo(A: Integer = 0)	YES	YES	TB.Foo(A: Integer = 0)
21	Foo(A: Integer)		1	NO			NO		
22	Foo(A: Integer)		2	YES	YES	TA.Foo(A: Integer = 0)	NO		
23	Foo(A: Integer)		3	YES	YES	TA.Foo(A: string = "")	NO		
24	Foo(A: Integer)		4	YES	NO	T0.Foo	YES	NO	T0.Foo
25		Foo(A: string = "")	1	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
26		Foo(A: string = "")	2	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
27		Foo(A: string = "")	3	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
28		Foo(A: string = "")	4	YES	YES	TB.Foo(A: string = "")	YES	YES	TB.Foo(A: string = "")
29		Foo(A: string)	1	NO			NO		
30		Foo(A: string)	2	YES	YES	TA.Foo(A: Integer = 0)	NO		
31		Foo(A: string)	3	YES	YES	TA.Foo(A: string = "")	NO		
32		Foo(A: string)	4	YES	NO	T0.Foo	YES	NO	T0.Foo