	static{}ClassMainApp2	Because "main" method is in Class MainApp2 and this class is the first to load to HEAP. That is why static module{} is the first to execute. In static{}ClassMainApp2 we are trying to create instance of Class C whose ancestry are Class B and A. That is why we should first go to Class A and see what is there. Then we will visit Class B. And Class C at the end. After we will get back here to create instance of C. ВНИМАНИЕ!!! Чтобы понять всю логику нужно сначала пробежаться по всей цепочке от С к А. Приходим в С. Так как С наследуется от В, то в классе С ничего не делаем. Идем в В и там пока ничего не делаем, т.к. В наследуется от А. Идем в А. Дальше идти некуда(Точнее идем в Object и возвращаемся обратно в А). Загружаем информацию о классе А в экземпляр класса Class. Инициализируем поля класса А, выполняем статические блоки класса А. Если бы Класс С не наследовался от В, то необходимости бегать по предкам не было бы. Мы сразу оказались бы в цепочке: new C() → invoke C() → invoke Object()
	static{}ClassA	Module static{}Class A is first to execute. There is nothing to execute here but sysout. Let's go to B.
	static{}ClassB	Module static{}Class B is first to execute. Oh, we should create instance of Class B here. Class B is an ancestor of A. But I'm just from A and everything is done there. We can create instance of B. Let's see: new B() → invoke B() → invoke B(int) → invoke A(int) → invoke A() → invoke A(String) → invoke Object()
		The output of all this is:
	{}ClassA	Before execute the A(String) we shold execute {}ClassA Внимание!!! Создание полей экземляра класса и выполнение блоков (не статических) происходит каждый раз перед вызовом конструктора .
	A(String)	
	A()	
	A(int)	
	{1}ClassB	Before execute the B(int) we shold execute {1}ClassB and {2}ClassB
	{2}ClassB	
	B(int)	
	B()	Done with new B() from static{}ClassB.
	static{}ClassC	Let't see what is in Class C. Here we have static{}ClassC to execute. And nothing more.
		Go back to static{}ClassMainApp2 and create instance of Class C. new C()-->invoke C()--> invoke B()-->invoke B(int)-->invoke A(int)-->invoke A() -->invoke A(String)-->invoke Object()
	{}ClassA	Before execute the A(String) we shold execute {}ClassA
	A()	
	A(int)	
	{1}ClassB	Before execute the B(int) we shold execute {1}ClassB and {2}ClassB
	{2}ClassB	
	B(int)	
	B()	
	{}ClassC	Before execute the C() we shold execute {}ClassC
	C()	Now, we are done with static{}ClassMainApp2. Let's go to “main”

	main (String[] args)	
		Stream 1 In the “main ” we shoul create instance of Class C with parametr new B. new C (new B()) → invoke C (Object) → invoke C(String) → invoke B() → invoke B(int) → invoke A(int) → invoke A() → invoke A(String) → invoke Object() Stream 2 new B() → invoke B() → invoke B(int) → invoke A(int) → invoke A() → invoke A(String) → invoke Object()
	{}ClassA	Stream 2. Before execute the A(String) we shold execute {}ClassA
	A(String)	
	A()	
	A(int)	
	{1}ClassB	Before execute the B(int) we shold execute {1}ClassB and {2}ClassB
	{2}ClassB	
	B(int)	
	B()	
	{}ClassA	Stream 1.Before execute the A(String) we shold execute {}ClassA
	A(String)	
	A()	
	A(int)	
	{1}ClassB	Before execute the B(int) we shold execute {1}ClassB and {2}ClassB
	{2}ClassB	
	B(int)	
	B()	
	{}ClassC	Before execute the C(String) we shold execute {}ClassC
	C(String)	
	C (Object)	
	The END!!!	