

<i>termvar</i> , <i>x</i> , <i>y</i> , <i>z</i> , <i>f</i> <i>modName</i> , <i>M</i> <i>i</i> , <i>j</i>		
<i>prog</i>	$::=$ <i>topMod</i> ₁ ... <i>topMod</i> _{<i>i</i>}	Program
<i>topMod</i>	$::=$ module <i>M</i> ₁ : { <i>modSig</i> } module <i>M</i> ₂ { <i>modBody</i> } <i>mod</i> module <i>M</i> ₁ : { <i>modSig</i> }	Top level module
<i>mod</i>	$::=$ module <i>M</i> { <i>modBody</i> } module <i>M</i> (<i>x</i> ₁ : <i>M</i> ₁ , ... , <i>x</i> _{<i>i</i>} : <i>M</i> _{<i>i</i>}){ <i>modBody</i> } module <i>M</i> = <i>M</i> ₁ (<i>M</i> ₂)	Modules
<i>modBody</i>	$::=$ <i>importExp</i> ₁ ... <i>importExp</i> _{<i>i</i>} <i>topDef</i> ₁ ... <i>topDef</i> _{<i>j</i>} <i>topDef</i> ₁ ... <i>topDef</i> _{<i>i</i>}	Module body Import a module Top level definition.
<i>importExp</i>	$::=$ import <i>x</i> (<i>y</i> ₁ , ..., <i>y</i> _{<i>i</i>}) import <i>x</i> (<i>y</i> ₁ , ..., <i>y</i> _{<i>i</i>}) hiding (<i>z</i> ₁ , ... , <i>z</i> _{<i>j</i>})	Import Expression
<i>topDef</i>	$::=$ struct <i>x</i> {} class <i>x</i> {} <i>funDef</i>	Top level definitions Structs Classes Function Definitions
<i>funDef</i>	$::=$ <i>x</i> ₁ : <i>funType</i> <i>x</i> ₂ <i>funParams</i> = t <i>x</i> ₁ : <i>funType</i> <i>x</i> ₂ (<i>funParams</i>){ t }	Top-level function definitions Functional Style Swift Style
<i>funType</i>	$::=$ <i>T</i> ₁ → ... → <i>T</i> _{<i>i</i>}	Function Types
<i>funParams</i>	$::=$ <i>y</i> ₁ , ... , <i>y</i> _{<i>i</i>}	Function Parameters
<i>modSig</i>	$::=$	Module Signature
<i>K</i>	$::=$	Kinds
<i>T</i>	$::=$?	Types The dynamic type.