

<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>	$::=$ $  \quad topMod_1 \dots topMod_i$	Program
<i>topMod</i>	$::=$ $  \quad \text{module } M_1 : \{modSig\} \text{ module } M_2 \{modBody\}$ $  \quad \text{mod}$ $  \quad \text{module } M_1 : \{modSig\}$	Top level module
<i>mod</i>	$::=$ $  \quad \text{module } M \{modBody\}$ $  \quad \text{module } M(x_1 : M_1, \dots, x_i : M_i) \{modBody\}$ $  \quad \text{module } M = M_1(M_2)$	Modules
<i>modBody</i>	$::=$ $  \quad importExp_1 \dots importExp_i topDef_1 \dots topDef_j$ $  \quad topDef_1 \dots topDef_i$	Module body Import a module Top level definition.
<i>importExp</i>	$::=$ $  \quad \text{import } x(y_1, \dots, y_i)$ $  \quad \text{import } x(y_1, \dots, y_i) \text{ hiding } (z_1, \dots, z_j)$	Import Expression
<i>topDef</i>	$::=$ $  \quad \text{struct } x \{ \}$ $  \quad \text{class } x \{ \}$ $  \quad funDef$	Top level definitions Structs Classes Function Definitions
<i>funDef</i>	$::=$ $  \quad x_1 : funType \ x_2 \ funParams = \mathbf{t}$ $  \quad x_1 : funType \ x_2 (funParams) \{ \mathbf{t} \}$	Top-level function definitions Functional Style Swift Style
<i>funType</i>	$::=$ $  \quad T_1 \rightarrow \dots \rightarrow T_i$	Function Types
<i>funParams</i>	$::=$ $  \quad y_1, \dots, y_i$	Function Parameters
<i>modSig</i>	$::=$	Module Signature
<i>K</i>	$::=$	Kinds
<i>T</i>	$::=$ $  \quad ?$	Types The dynamic type.