```
class Cat {
	public int age;
	private int weight;
	public void run(int x, int y) {
		int speed;
		int arr[];
		…
		speed = 7;
	}
	Cat() {}
}

void kxi2017 main() {
	char c
	…
	c = 's';
}
```

Symbol Table …

Attributes

Scope – region of a program where an identifier is accessible
Symbol ID – a unique identifier
Value - Lexeme
Kind – is the type of the Symbol
Data – kind specific data

C100 →	Scope:	g.
	Symid:	C100
	Value:	Cat
	Kind:	Class
	Data:	
V101 →	Scope:	g.Cat
	Symid:	V101
	Value:	age
	Kind:	ivar /* instance variable *
	Data:	type: int
		accessMod: public
V102 →	Scope:	g.Cat
	Symid:	V102
	Value:	weight
	Kind:	ivar // instance variable
	Data:	type: int
		accessMod: private
M103 →	Scope:	g.Cat
	Symid:	M103
	Value:	run
	Kind:	method
	Data:	returnType: void
		Param: [P104, P105]
		accessMod: public

P104 → Scope: g.Cat.run
Symid: P104
Value: x
Kind: param
Data: type: int
accessMod: private

P105 → Scope: g.Cat.run
Symid: P105
Value: y
Kind: param
Data: type: int
accessMod: private

L106→ Scope: g.Cat.run
Symid: L106
Value: speed
Kind: lvar // local variable
Data: type: int
accessMod: private

L107→ Scope: g.Cat.run
Symid: L107
Value: arr
Kind: lvar // local variable
Data: type: @:int // Array of int
accessMod: private

X108 → Scope: g.Cat
Symid: X108
Value: Cat
Kind: Constructor
Data: returnType: Cat
Param: []
accessMod: public

F110 → Scope: g
Symid: F110
Value: main
Kind: main
Data: returnType: void
Param: []
accessMod: public

N118 → Scope: g
Symid: N108
Value: 7
Kind: ilit // integer literal
Data: type: int
accessMod: public

H121 → Scope: g
Symid: H108
Value: 's'
Kind: clit // character literal
Data: type: char
accessMod: public