DFINITY

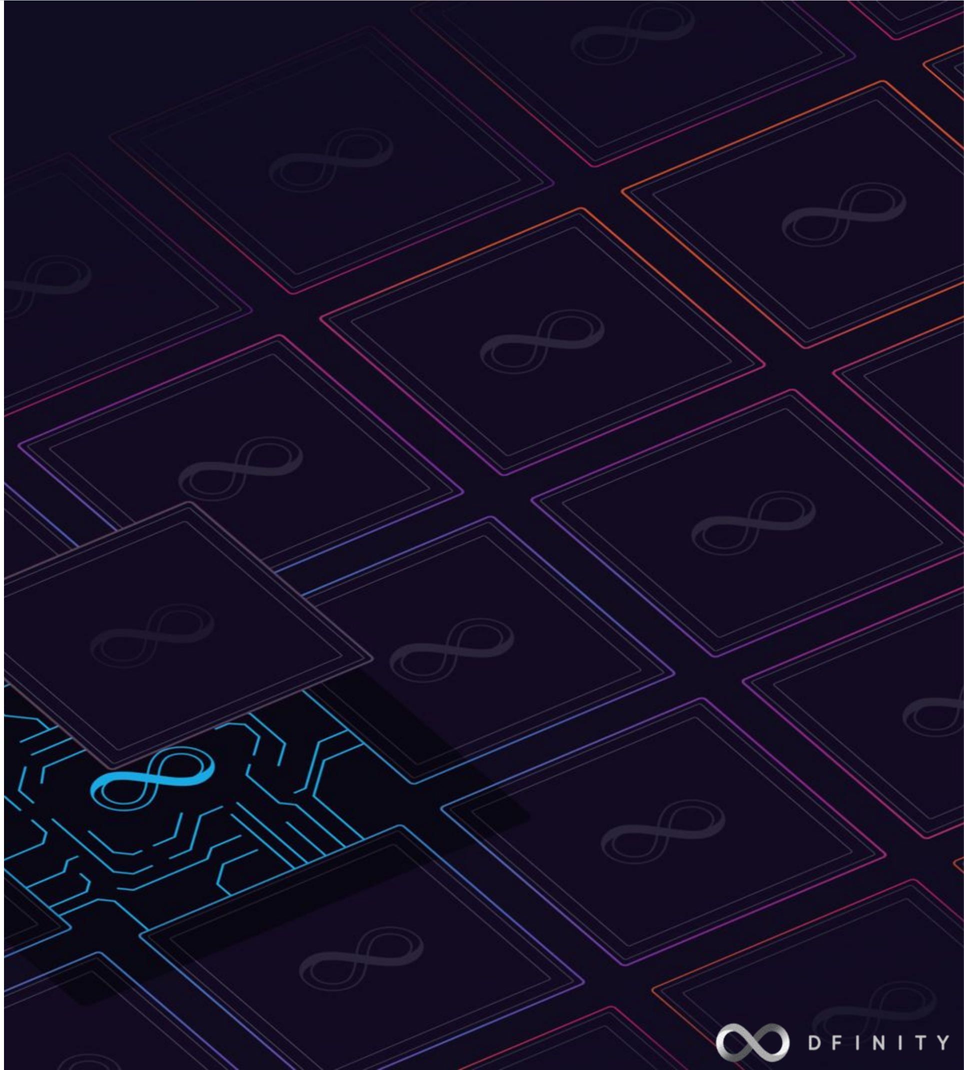
ICP区块链开发进阶课程

1. Motoko 语言进阶

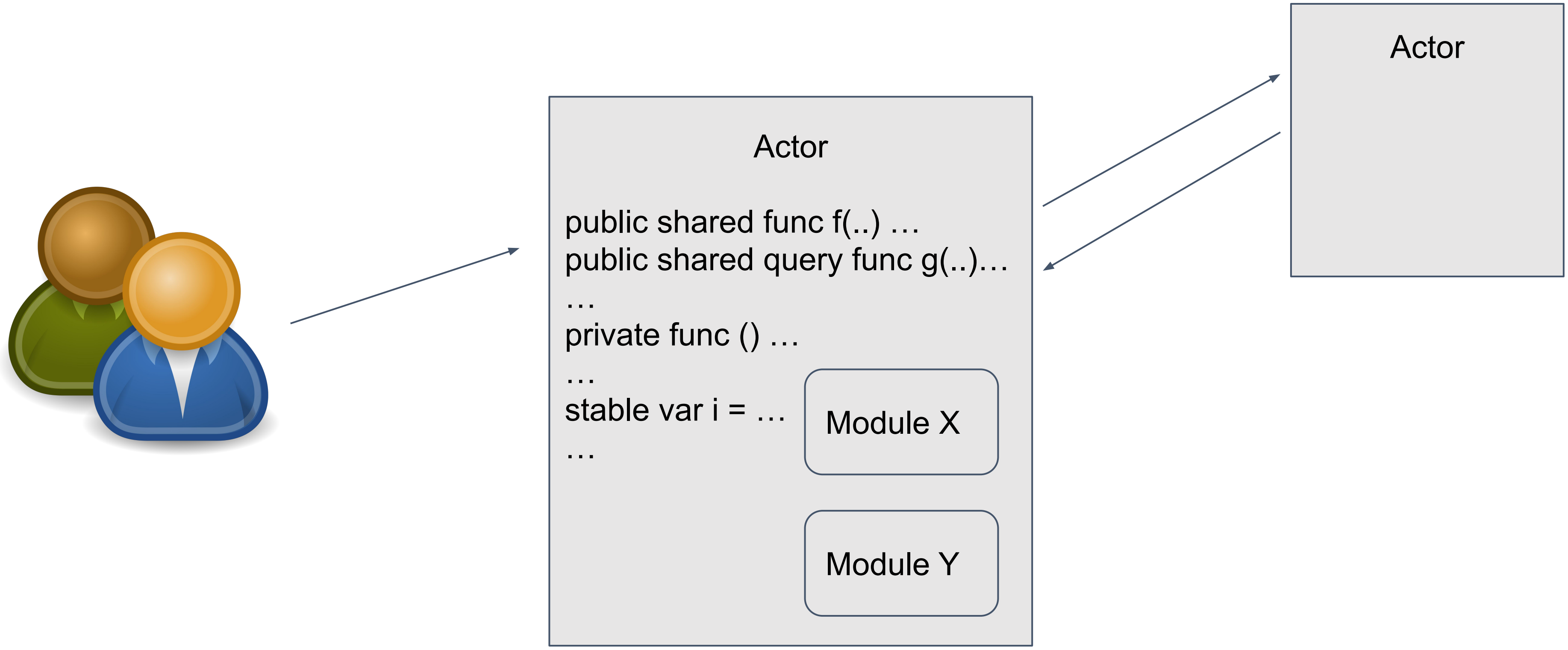
主讲：Paul Liu - DFINITY 工程师

课程大纲

1. Motoko 语言进阶
2. Canister 开发进阶 I
3. Canister 开发进阶 II
4. 整合 ICP 系统服务
5. 项目实例分析



Motoko Canister 容器



Actor

public shared func f(..) …
public shared query func g(..)…
…
private func () …
…
stable var i = …
…

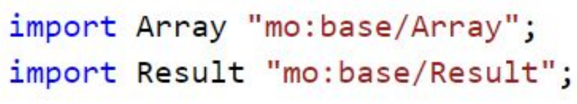
Module X

Module Y

Actor

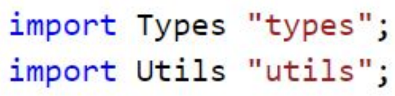
Module import

- 库模块



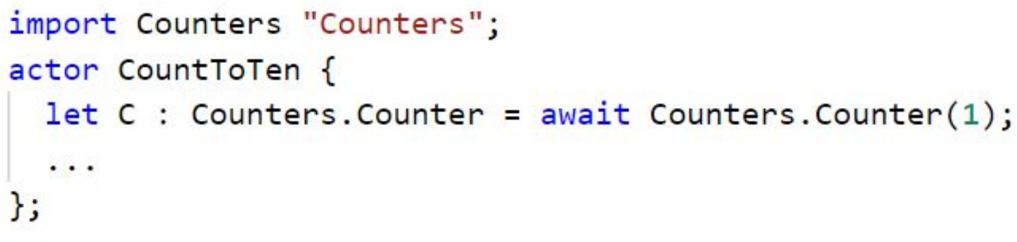
```
import Array "mo:base/Array";
import Result "mo:base/Result";
```

- 本地模块



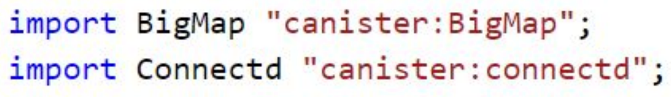
```
import Types "types";
import Utils "utils";
```

- Actor Class



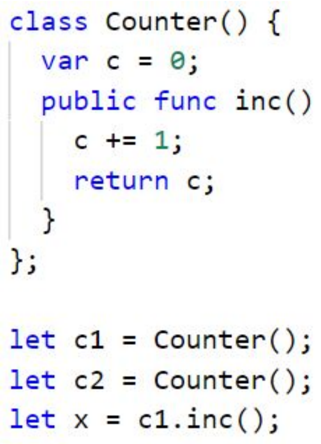
```
import Counters "Counters";
actor CountToTen {
  let C : Counters.Counter = await Counters.Counter(1);
  ...
};
```

- Canister

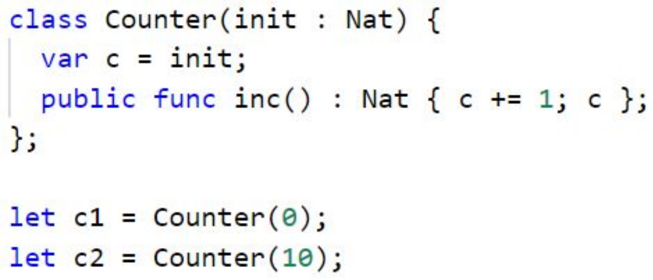


```
import BigMap "canister:BigMap";
import Connectd "canister:connectd";
```

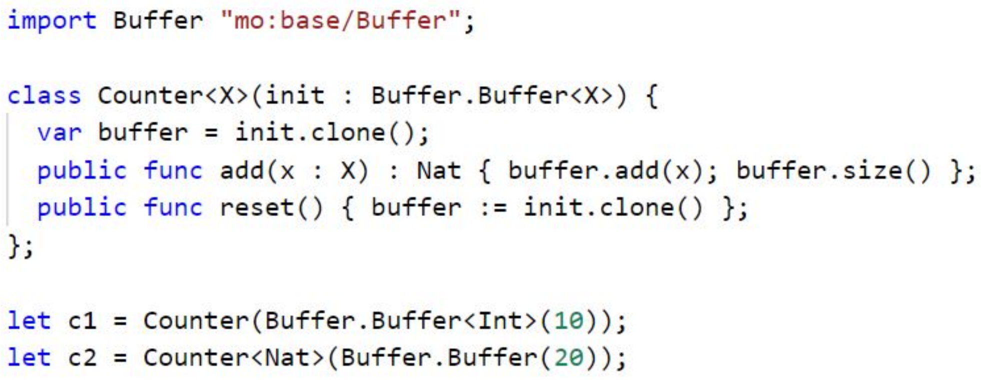
Object 和 Class



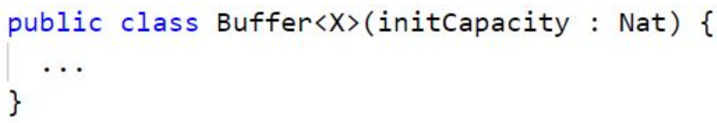
```
class Counter() {
  var c = 0;
  public func inc() : Nat {
    c += 1;
    return c;
  }
};

let c1 = Counter();
let c2 = Counter();
let x = c1.inc();
```



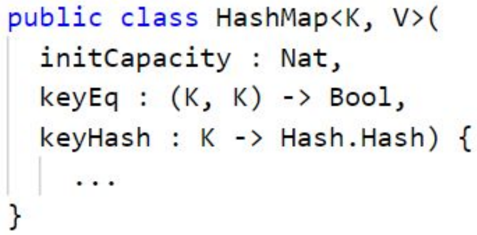
```
class Counter(init : Nat) {
  var c = init;
  public func inc() : Nat { c += 1; c };
};

let c1 = Counter(0);
let c2 = Counter(10);
```



```
import Buffer "mo:base/Buffer";

class Counter<X>(init : Buffer.Buffer<X>) {
  var buffer = init.clone();
  public func add(x : X) : Nat { buffer.add(x); buffer.size() };
  public func reset() { buffer := init.clone() };
};

let c1 = Counter(Buffer.Buffer<Int>(10));
let c2 = Counter<Nat>(Buffer.Buffer(20));
```

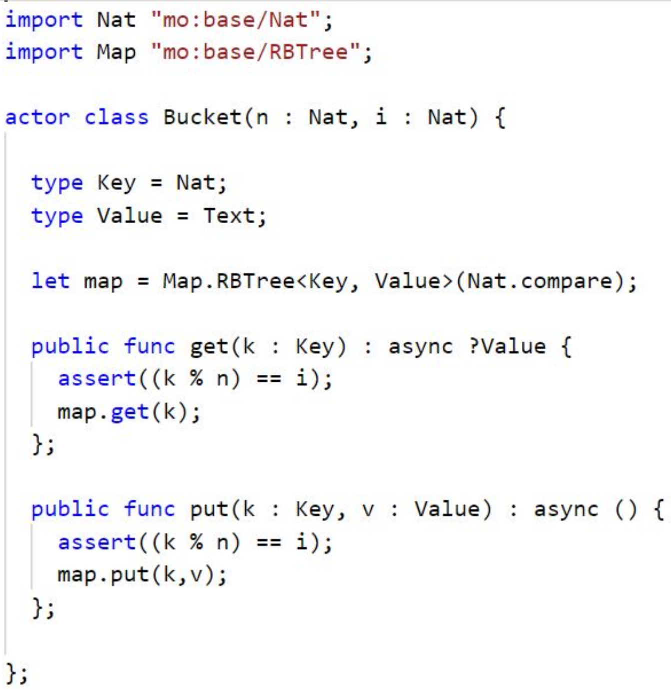


```
public class Buffer<X>(initCapacity : Nat) {
  ...
}
```

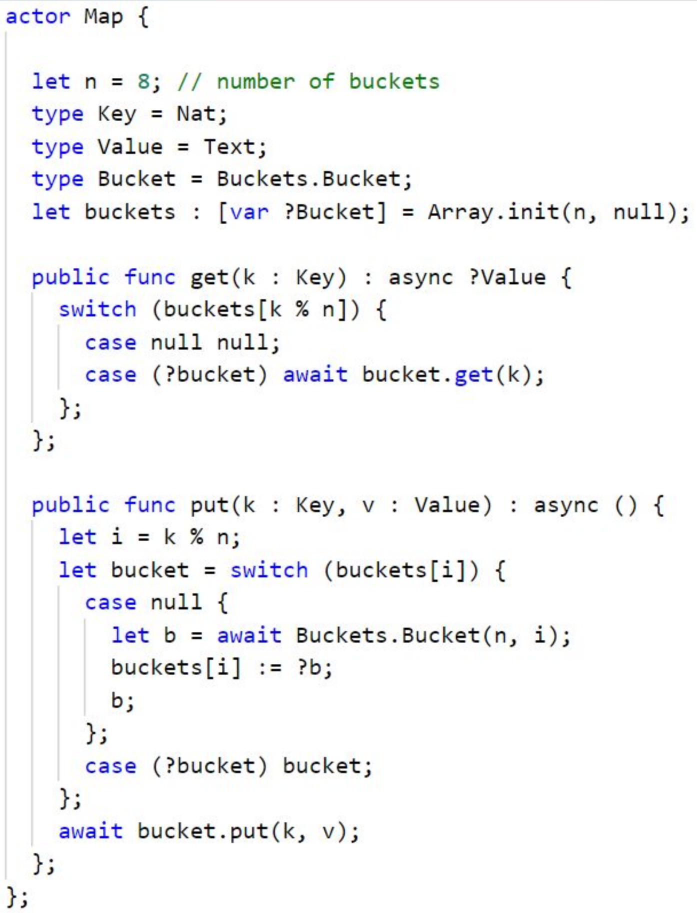


```
public class HashMap<K, V>(
  initCapacity : Nat,
  keyEq : (K, K) -> Bool,
  keyHash : K -> Hash.Hash) {
    ...
}
```

Actor Class



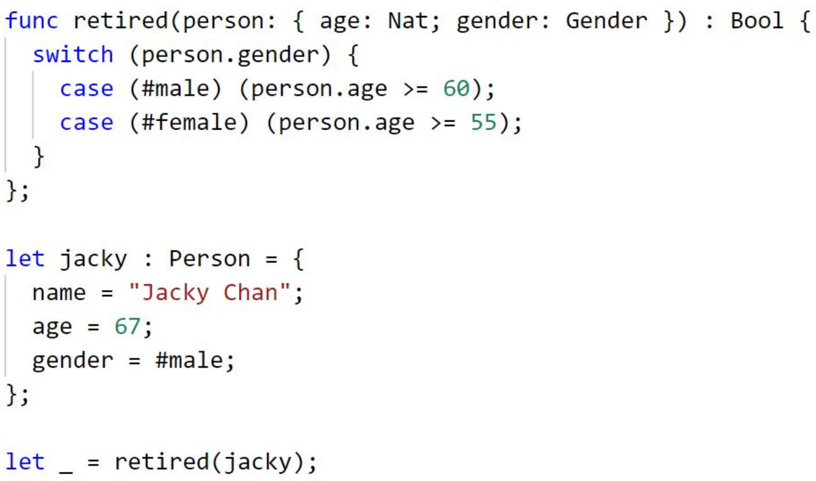
```
import Nat "mo:base/Nat";
import Map "mo:base/RBTree";

actor class Bucket(n : Nat, i : Nat) {

  type Key = Nat;
  type Value = Text;

  let map = Map.RBTree<Key, Value>(Nat.compare);

  public func get(k : Key) : async ?Value {
    assert((k % n) == i);
    map.get(k);
  };

  public func put(k : Key, v : Value) : async () {
    assert((k % n) == i);
    map.put(k,v);
  };

};
```



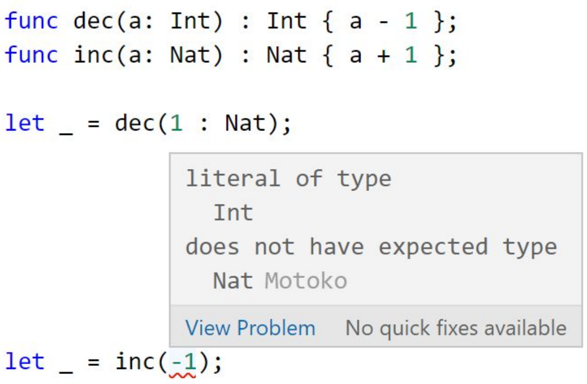
```
actor Map {

  let n = 8; // number of buckets
  type Key = Nat;
  type Value = Text;
  type Bucket = Buckets.Bucket;
  let buckets : [var ?Bucket] = Array.init(n, null);

  public func get(k : Key) : async ?Value {
    switch (buckets[k % n]) {
      case null null;
      case (?bucket) await bucket.get(k);
    };
  };

  public func put(k : Key, v : Value) : async () {
    let i = k % n;
    let bucket = switch (buckets[i]) {
      case null {
        let b = await Buckets.Bucket(n, i);
        buckets[i] := ?b;
        b;
      };
      case (?bucket) bucket;
    };
    await bucket.put(k, v);
  };
};
```

子类型关系 (subtype)

- B ≤ A
- B 是 A 的子类型
- 所有接受 A 类型的地方都可以用 B 类型的值
- A 更宽泛 (general)，B 更具体 (specific)



```
func retired(person: { age: Nat; gender: Gender }) : Bool {
  switch (person.gender) {
    case (#male) (person.age >= 60);
    case (#female) (person.age >= 55);
  }
};

let jacky : Person = {
  name = "Jacky Chan";
  age = 67;
  gender = #male;
};

let _ = retired(jacky);
```



```
func dec(a: Int) : Int { a - 1 };
func inc(a: Nat) : Nat { a + 1 };

let _ = dec(1 : Nat);
```

```
literal of type
  Int
does not have expected type
  Nat Motoko
```

```
let _ = inc(-1);
```

使用 vessel 管理程序库

1. 安装 vessel 命令行 https://github.com/dfinity/vessel

2. 初始化项目文件 `vessel init`

3. 修改 `vessel.dhall` 设置，并安装程序库 `vessel install`

```
{
  dependencies = [ "base", "matchers", "sha256"],
  compiler = None Text
}
```

4. 修改 `dfx.json` 设置：

```
{
  "defaults": {
     "build": { "packtool": "vessel sources" }
  }
}
```

5. 在 Motoko 程序中引用，比如 `import sha256 "mo:sha256/SHA256";`

单元测试

使用 matchers 库建立单元测试 https://github.com/kritzcreek/motoko-matchers

```
import Matchers "mo:matchers/Matchers";
import T "mo:matchers/Testable";
import Suite "mo:matchers/Suite";

let equals10 = Matchers.equals(T.nat(10));
let equals20 = Matchers.equals(T.nat(20));

let suite = Suite.suite("Testing the testing", [
    Suite.suite("equality", [
        Suite.test("nats1", 10, equals10),
        Suite.test("nats2", 20, equals10),
        Suite.test("Chars", 'a', Matchers.equals(T.char('b'))),
    ]),
];

Suite.run(suite)
```

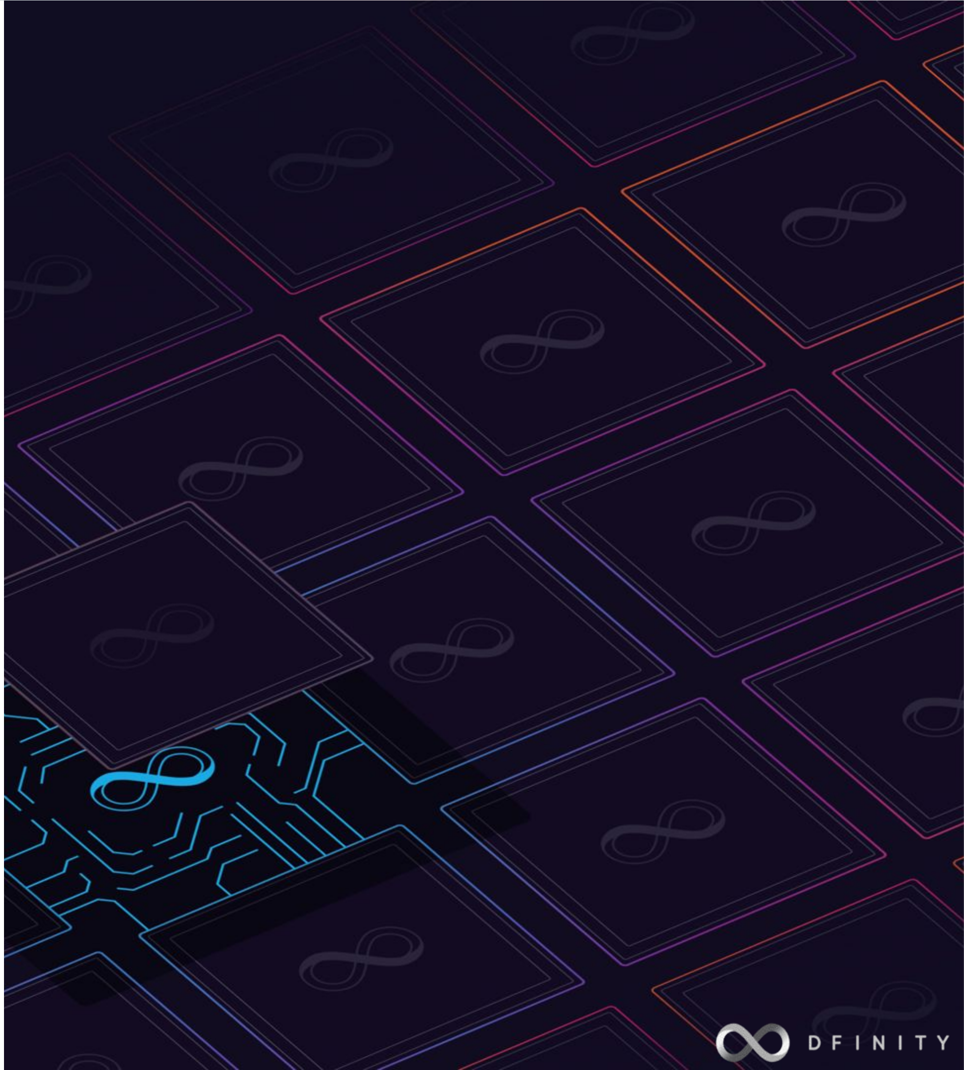
运行单元测试 `$(dfx cache show)/moc $(vessel sources) -wasi-system-api -r Test.mo`

Logger 演示

课程作业 1

判断下述子类型关系是否为真

```
{a: Bool} ≤ {a: Bool; b: Nat}

{a: Bool} ≤ {}

{#red; #blue} ≤ {#red; #yellow; #blue}

Nat ≤ Int

Int ≤ Int32

() -> () ≤ (Text) -> ()

() -> (Text) ≤ () -> ()

() -> ({#male; #female}) ≤ () -> ()

(Int) -> (Nat) ≤ (Nat) -> (Int)

(Int16, Nat8) ≤ (Int32, Nat32)
```



课程作业 2

实现一个可以无限扩容的 logger。

1. 假定一个 logger 装载 N 条信息之后就满了
2. 当一个 logger 容器满了之后，自动生成一个新的 logger 容器承载之后的消息

下一节：Canister 开发进阶 I

- 调用系统 API
- Candid 接口描述语言
- 错误及异常处理

DFINITY