

class System {

Producer p;

Consumer c;

FIFO fifo;

11.class IPull {

14.class IPush {

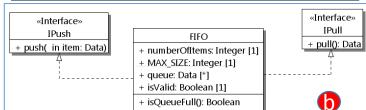
15.public:

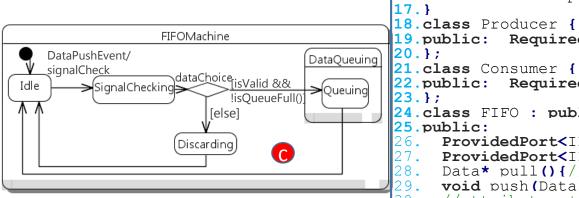
2. public:

9.

10.}

13.1





```
31. Statemachine FIFOMachine {
                                              InitialState Idle;
                                         33.
                                              State SignalChecking {
                                         34.
                                                StateEntry entryCheck;
                                         35.
                                                StateExit exitCheck;
    void configuration(){
                                         36.
                                             };
      bindPorts(p.pPush, fifo.pPush);
                                         37.
                                              State DataQueuing {
      bindPorts(c.pPull, fifo.pPull);
                                         38.
                                                StateEntry entryQueue;
                                         39.
                                                State Oueuing;
                                         40.
                                              };
                                         41.
                                              State Discarding;
12.public: virtual Data* pull() = 0;
                                              PseudoChoice dataChoice;
                                         43.
                                              CallEvent(DataPushEvent, push (Data&));
                                         44.
                                              TransitionTable {
                                         45.
                                                ExT(Idle, SignalChecking,
16. virtual void push (Data& data) = 0; 46.
                                                     DataPushEvent, NULL, signalCheck);
                                                ExT (SignalChecking, dataChoice,
                                         47.
                                         48.
                                                              NULL, NULL, NULL);
19.public: RequiredPort<IPush> pPush;
                                         49.
                                                ExT (dataChoice, Queuing, NULL, valid, NULL)
                                         50. }
                                         51.};
22.public: RequiredPort<IPull> pPull;
                                         52.void entryCheck(){//fine-grained code}
                                         53.void exitCheck(){//fine-grained code}
24.class FIFO : public IPush, IPull {
                                         54.void entryError(){//fine-grained code}
                                         55.void signalCheck(Data& item) {
     ProvidedPort<IPush> pPush;
                                         56.//trans effect from Idle to SignalChecking
     ProvidedPort<IPull> pPull;
     Data* pull(){//fine-grained code}
                                         58.bool valid() {return isValid&&isQueueFull()}
     void push(Data& data){//..}
                                         59.}
     //attributes + methods...
```