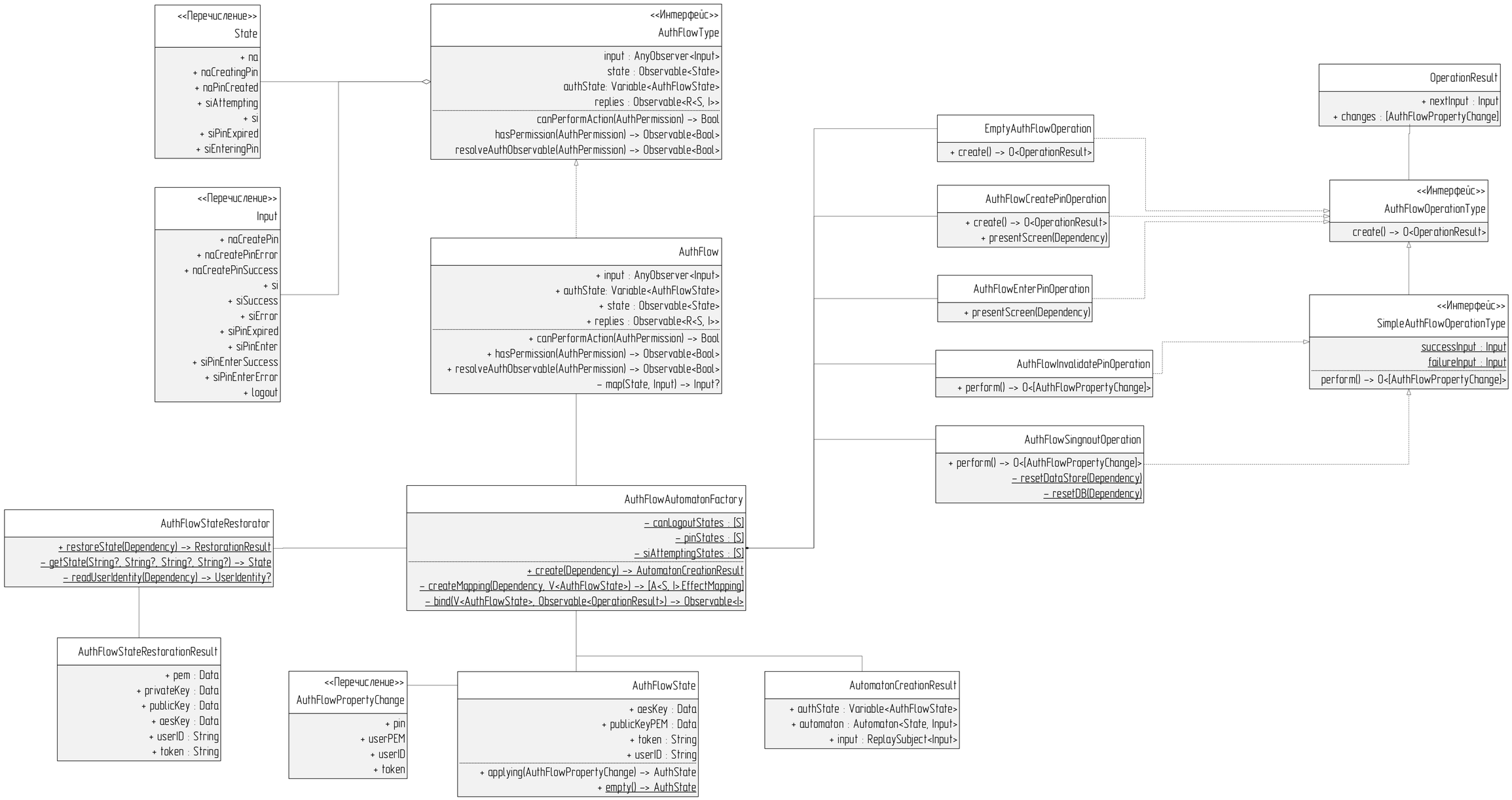
<<Перечисление>>
State
+ na
+ naCreatingPin
+ naPinCreated
+ siAttempting
+ si
+ siPinExpired
+ siEnteringPin
<<Перечисление>>
Input
+ naCreatePin
+ naCreatePinError
+ naCreatePinSuccess
+ si
+ siSuccess
+ siError
+ siPinExpired
+ siPinEnter
+ siPinEnterSuccess
+ siPinEnterError
+ logout
<<Интерфейс>>
AuthFlowType
input : AnyObserver<Input>
state : Observable<State>
authState: Variable<AuthFlowState>
replies : Observable<R<S, I>>
canPerformAction(AuthPermission) -> Bool
hasPermission(AuthPermission) -> Observable<Bool>
resolveAuthObservable(AuthPermission) -> Observable<Bool>
AuthFlow
+ input : AnyObserver<Input>
+ authState: Variable<AuthFlowState>
+ state : Observable<State>
+ replies : Observable<R<S, I>>
+ canPerformAction(AuthPermission) -> Bool
+ hasPermission(AuthPermission) -> Observable<Bool>
+ resolveAuthObservable(AuthPermission) -> Observable<Bool>
– map(State, Input) -> Input?
AuthFlowAutomatonFactory
– canLogoutStates : [S]
– pinStates : [S]
– siAttemptingStates : [S]
+ create(Dependency) -> AutomatonCreationResult
– createMapping(Dependency, V<AuthFlowState>) -> [A<S, I>.EffectMapping]
– bind(V<AuthFlowState>, Observable<OperationResult>) -> Observable<I>
AuthFlowStateRestorator
+ restoreState(Dependency) -> RestorationResult
– getState(String?, String?, String?, String?) -> State
– readUserIdentity(Dependency) -> UserIdentity?
AuthFlowStateRestorationResult
+ pem : Data
+ privateKey : Data
+ publicKey : Data
+ aesKey : Data
+ userID : String
+ token : String
<<Перечисление>>
AuthFlowPropertyChange
+ pin
+ userPEM
+ userID
+ token
AuthFlowState
+ aesKey : Data
+ publicKeyPEM : Data
+ token : String
+ userID : String
+ applying(AuthFlowPropertyChange) -> AuthState
+ empty() -> AuthState
AutomatonCreationResult
+ authState : Variable<AuthFlowState>
+ automaton : Automaton<State, Input>
+ input : ReplaySubject<Input>
EmptyAuthFlowOperation
+ create() -> O<OperationResult>
AuthFlowCreatePinOperation
+ create() -> O<OperationResult>
+ presentScreen(Dependency)
AuthFlowEnterPinOperation
+ presentScreen(Dependency)
AuthFlowInvalidatePinOperation
+ perform() -> O<[AuthFlowPropertyChange]>
AuthFlowSingnoutOperation
+ perform() -> O<[AuthFlowPropertyChange]>
– resetDataStore(Dependency)
– resetDB(Dependency)
OperationResult
+ nextInput : Input
+ changes : [AuthFlowPropertyChange]
<<Интерфейс>>
AuthFlowOperationType
create() -> O<OperationResult>
<<Интерфейс>>
SimpleAuthFlowOperationType
successInput : Input
failureInput : Input
perform() -> O<[AuthFlowPropertyChange]>