

## Template Method

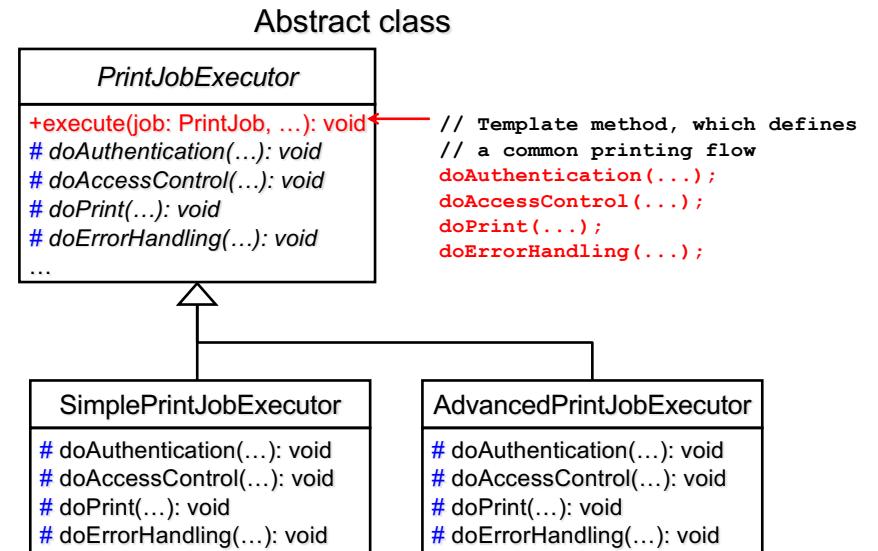
- Intent

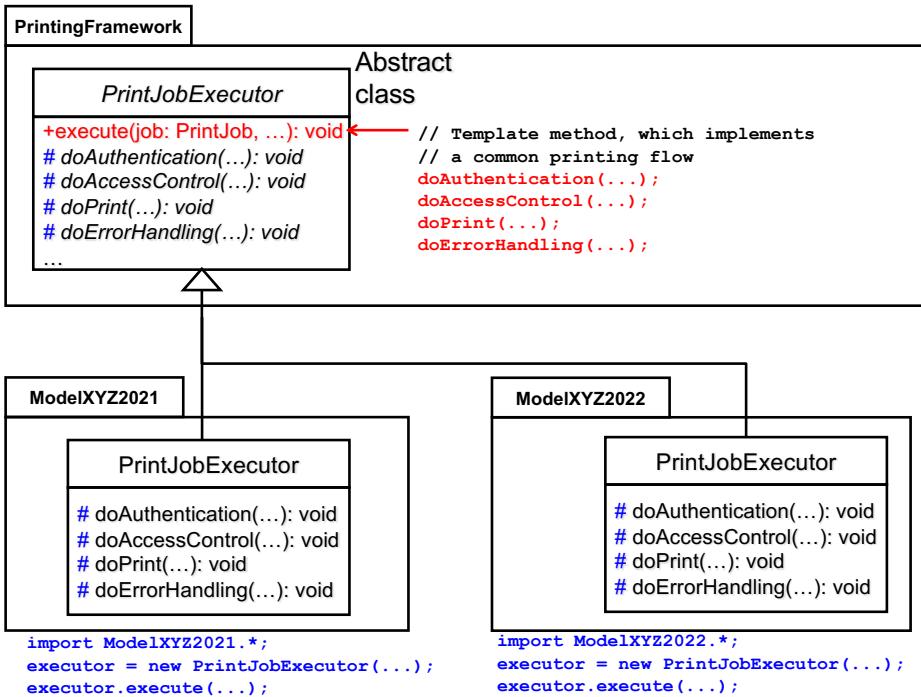
- Define a template of an algorithm (i.e., abstract flow of an algorithm) in a superclass's method
- Defer the implementation details of some steps in the algorithm to subclasses.
- Have subclasses override (or redefine) those steps without changing the algorithm's template/flow.
- Can reuse (or enforce) the template in all subclasses.

## Template Method Design Pattern

## Template for Printing Procedure

- Suppose you are implementing printer firmware
  - Common printing flow
    - Authentication
      - Enforce who can access the printer and who cannot.
    - Access control
      - Enforce who can use which services.
    - Print
    - Error handling
      - Manage what to do upon errors
  - Can define this common flow in a superclass and reuse the flow across its subclasses, which implement different firmware.
    - Different versions for the same printer model.
    - Different firmware for different printer models.





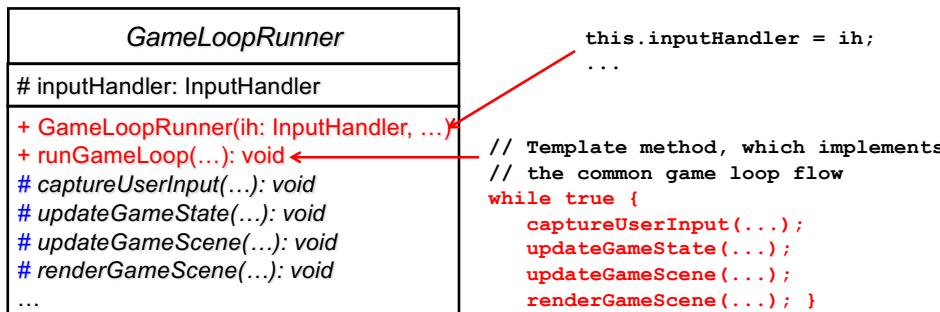
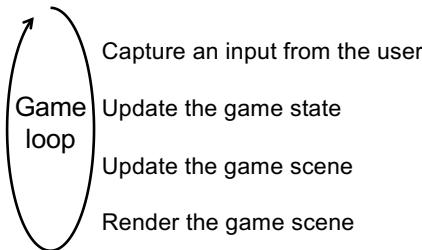
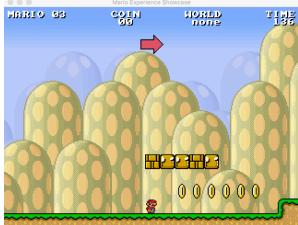
5

## Benefits

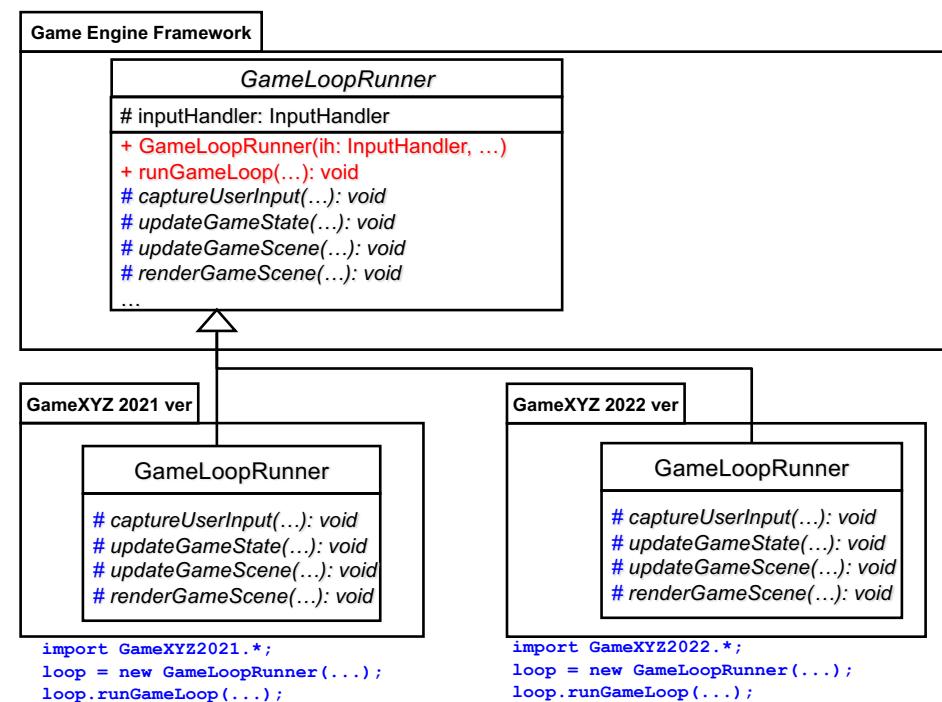
- Can explicitly define the flow of an algorithm.
  - Can separate individual steps in the flow clearly.
- Can consistently use/enforce the same flow across different subclasses
- Different subclasses can customize algorithm steps as necessary.

7

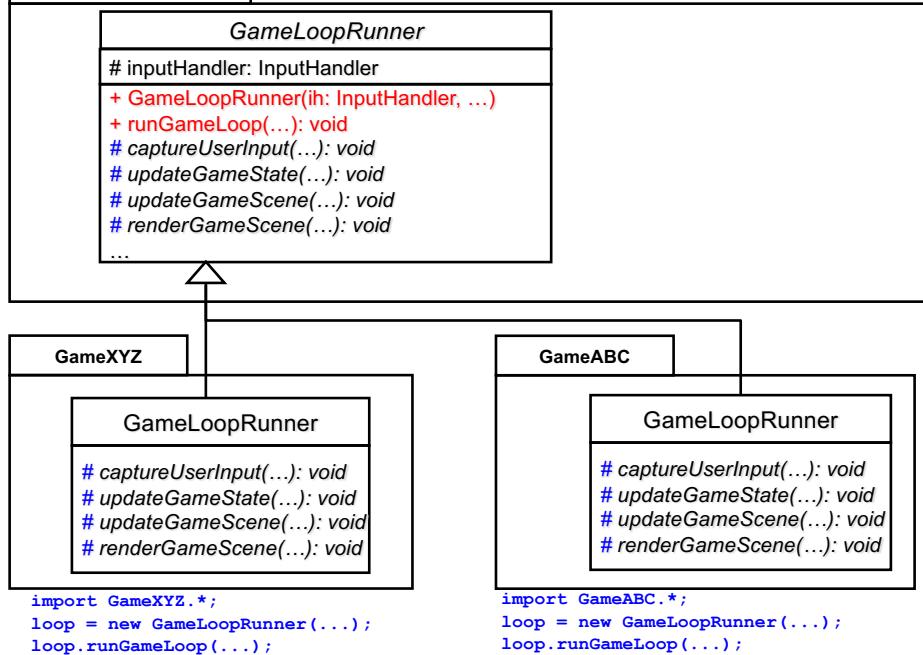
## Game Loop as a Template



8

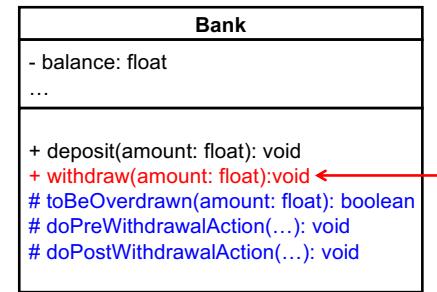


## Game Engine Framework



## A Template for Money Withdrawal in Banking

- There are many policies for a bank to deal with overdrafts.



```
if( toBeOverdrawn(amount) ){
    doPreWithdrawalAction(...);
} else{
    balance - amount;
    doPostWithdrawalAction(...);
}
```

## Other Template Examples

- Loan initiation (loan setup)
  - Credit score analysis
  - Transaction history analysis
  - Household analysis
  - etc. etc.
- Keyword search
  - Keyword analysis
  - Search
  - Search result ordering

