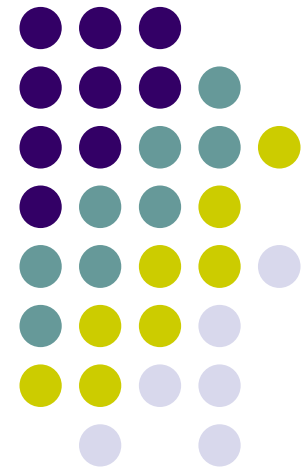


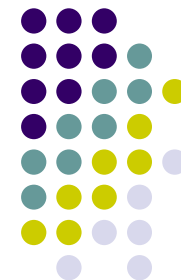
# UML



# UML



- **Unified Modeling Language** – унифицированный язык моделирования.
- **UML** представляет собой набор соглашений, которые предназначены для облегчения процесса моделирования и обмена информацией в проектной группе..



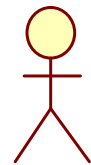
# Диаграммы

- Основу **UML** представляют диаграммы, которые различаются по типам и предназначены для моделирования различных аспектов разработки
- Типы диаграмм:
  - Поведенческие
  - Структурные
  - Развертывания

# Диаграммы вариантов использования (Use-case diagram)



- Actor (актор)



User

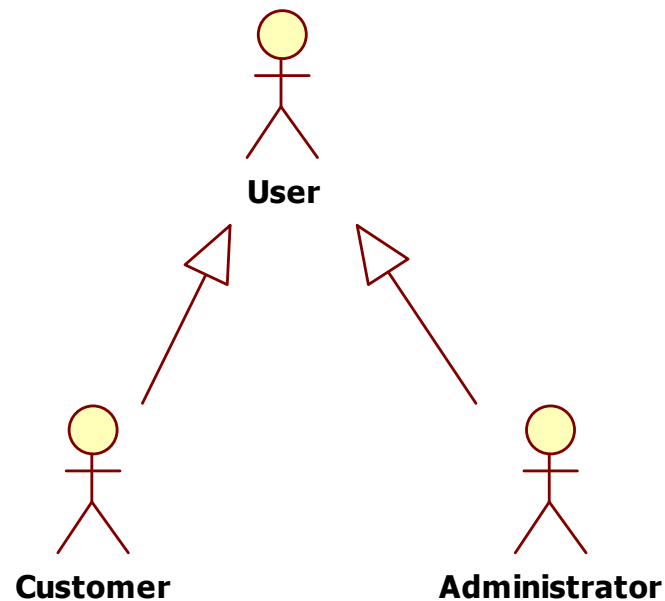
- Use-case (вариант использования)



# Диаграммы вариантов использования (Use-case diagram)



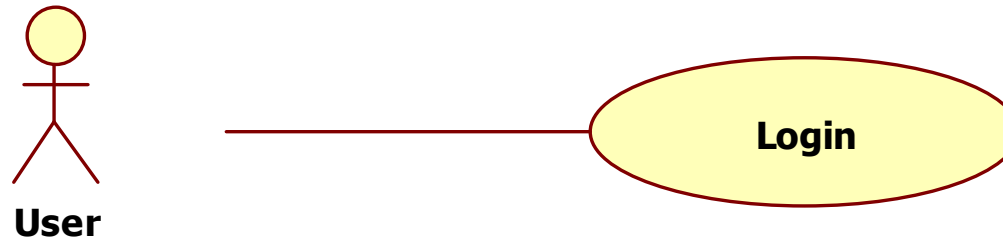
- Отношения
  - Наследование (генерализация)



# Диаграммы вариантов использования (Use-case diagram)



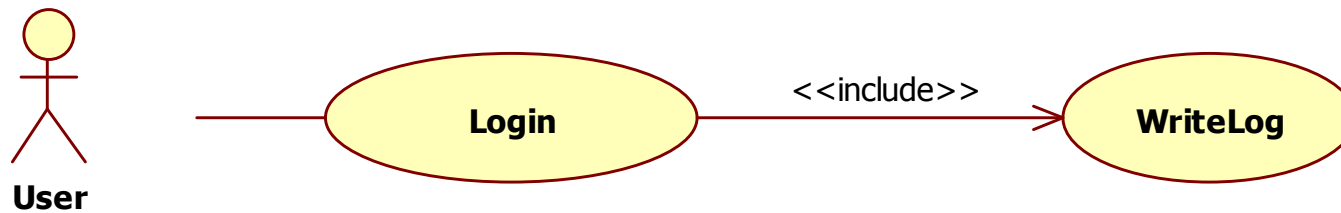
- Отношения
  - Ассоциация (инициирование варианта использования)



# Диаграммы вариантов использования (Use-case diagram)



- Отношения
  - Включение



# Диаграммы вариантов использования (Use-case diagram)



- Отношения
  - Расширение

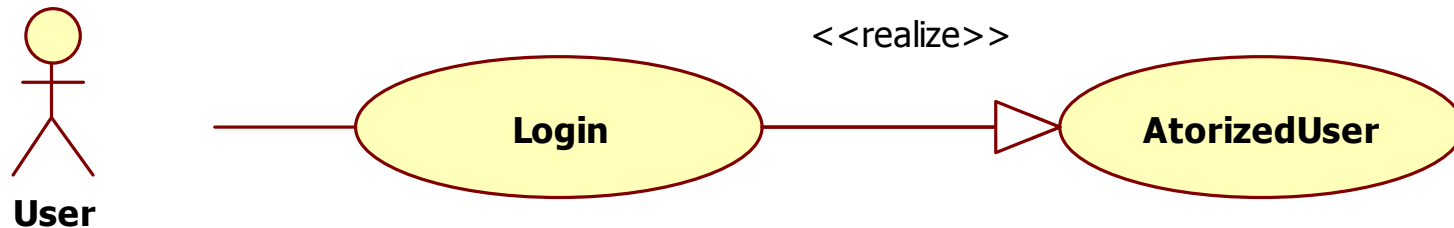


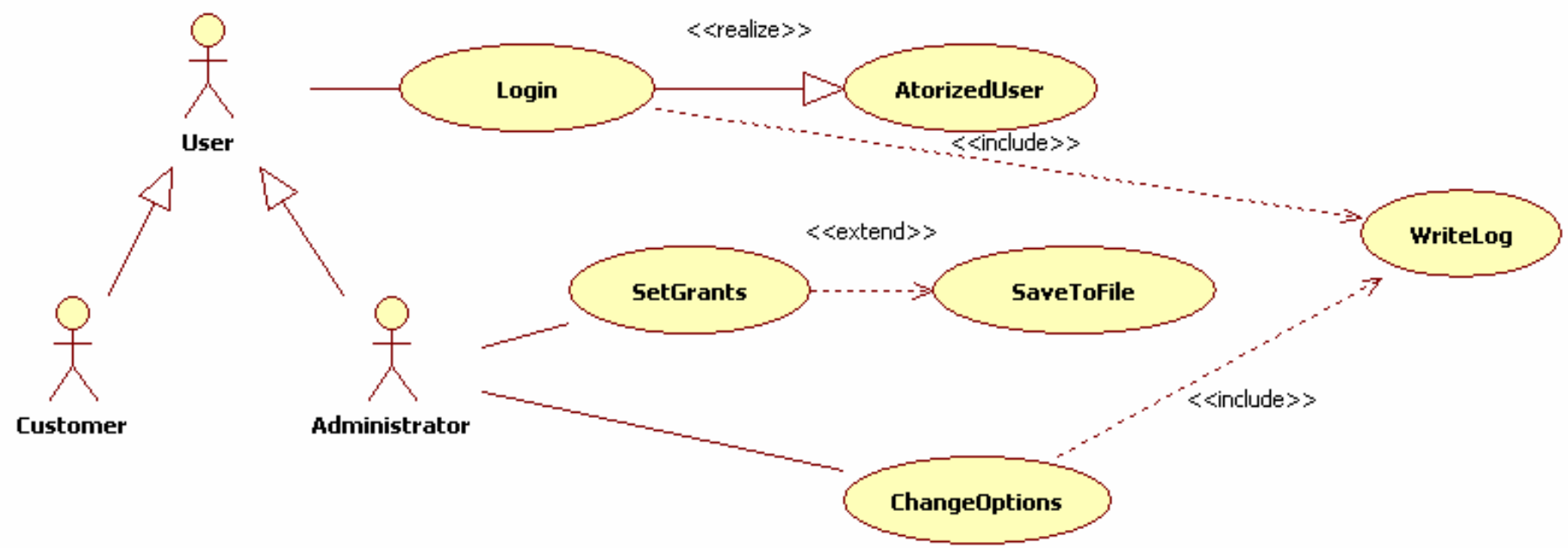
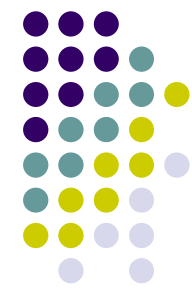


# Диаграммы вариантов использования (Use-case diagram)



- Отношения
  - реализация

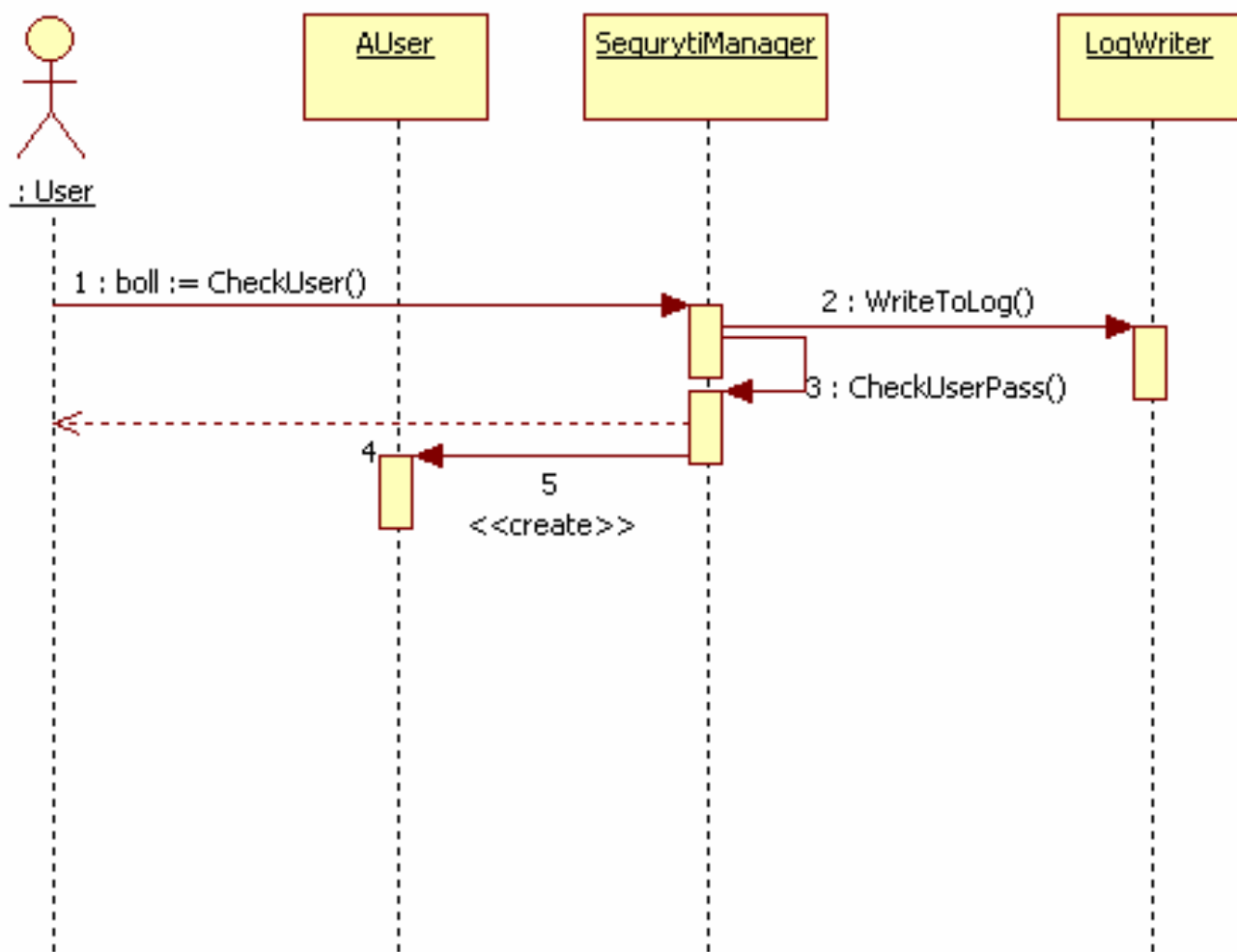


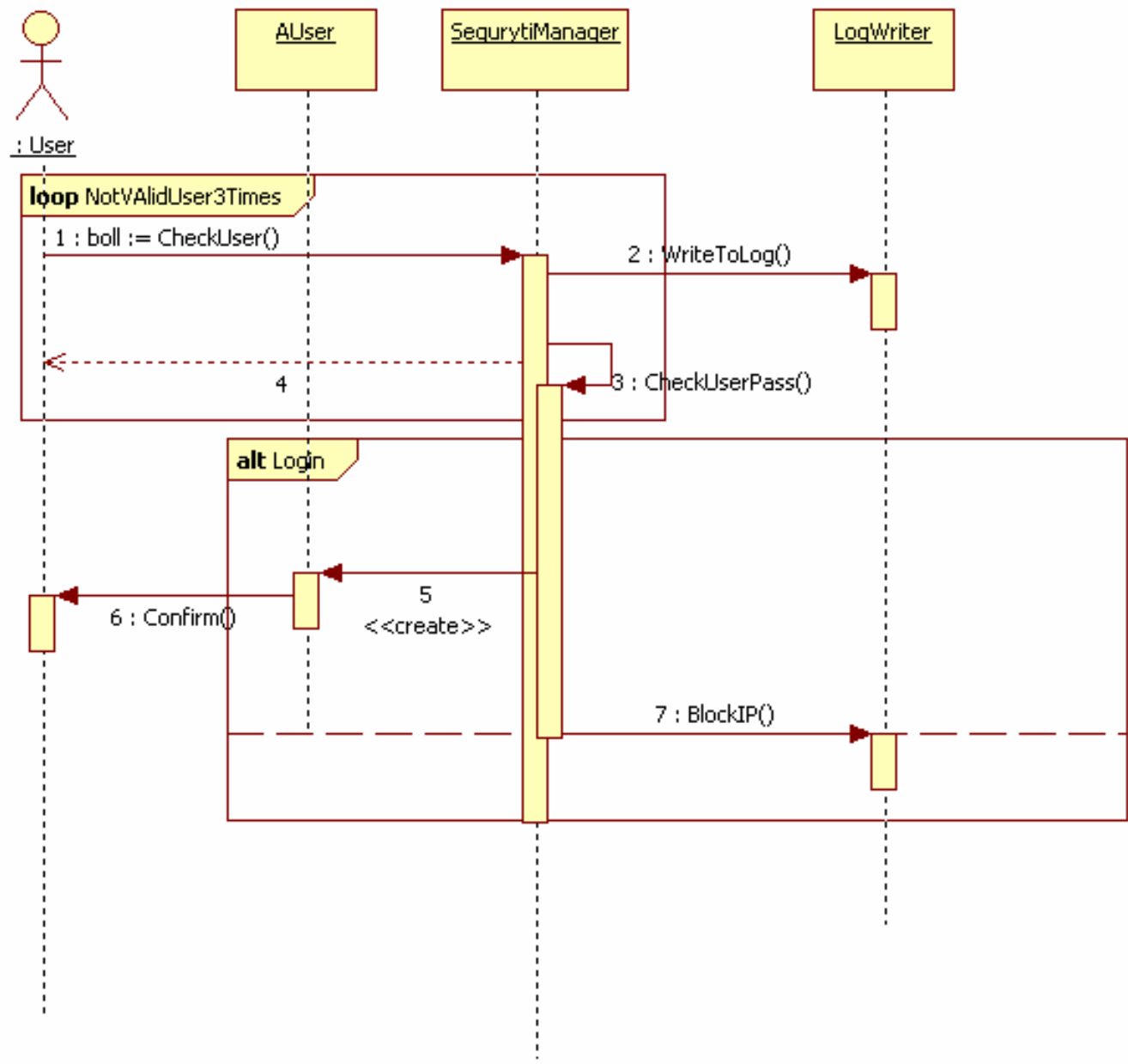


# Диаграммы последовательностей (Sequence diagram)



- Диаграмма последовательностей служит основным способом описания последовательности действий в процессе выполнения того или иного варианта использования.



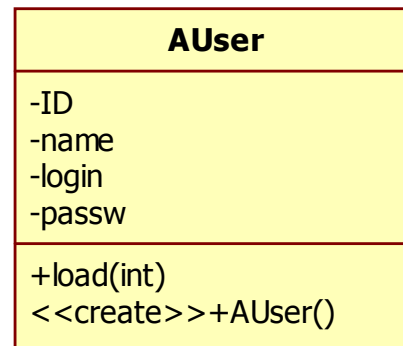




# Диаграммы классов

- Для отображения классов разрабатываемого приложения и их взаимосвязей
- Класс состоит из двух частей – заголовка с именем класса и тела с описанием его полей (**Атрибуты** – в терминах UML) и методов (**Операции** - в терминах UML).

# Диаграммы классов





# Атрибуты класса

- Атрибуты записываются с указанием доступности, имени и типа

<i><b>AUser</b></i>
-ID:int -name:string -login:string -passwd:string
+load(int) <<create>>+AUser()





# Операции (методы)

- Операции также записываются с указанием области видимости, имени и возвращаемого типа.
- Для них также указывается перечень принимаемых значений (параметров)

<i>AUser</i>
-ID -name -login -passw
+load(int) <<create>>+AUser()

# Операции (методы)

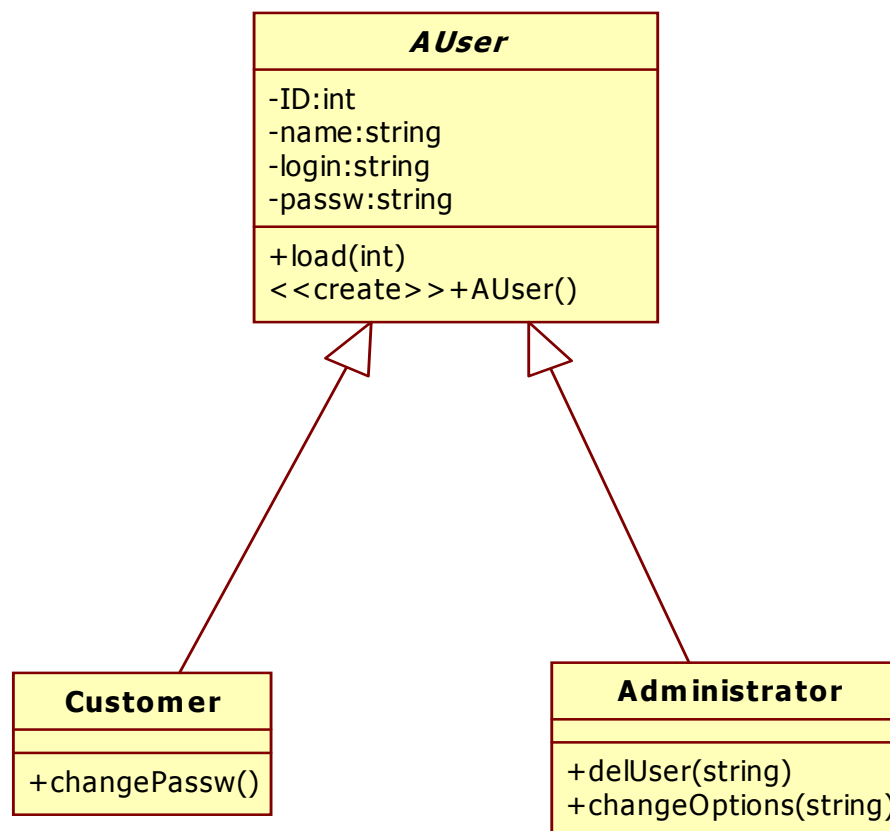


UserList
+addUser(int)
<u>+allUsers()</u>



# Взаимосвязи классов

- Генерализация или наследование (Generalize)





# Взаимосвязи классов

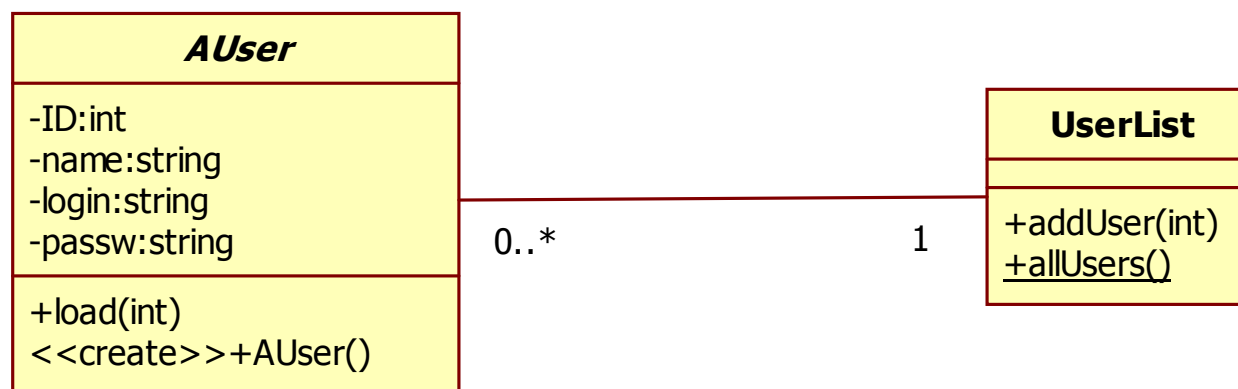
- Реализация (Realize)
- Данный класс реализует данный интерфейс



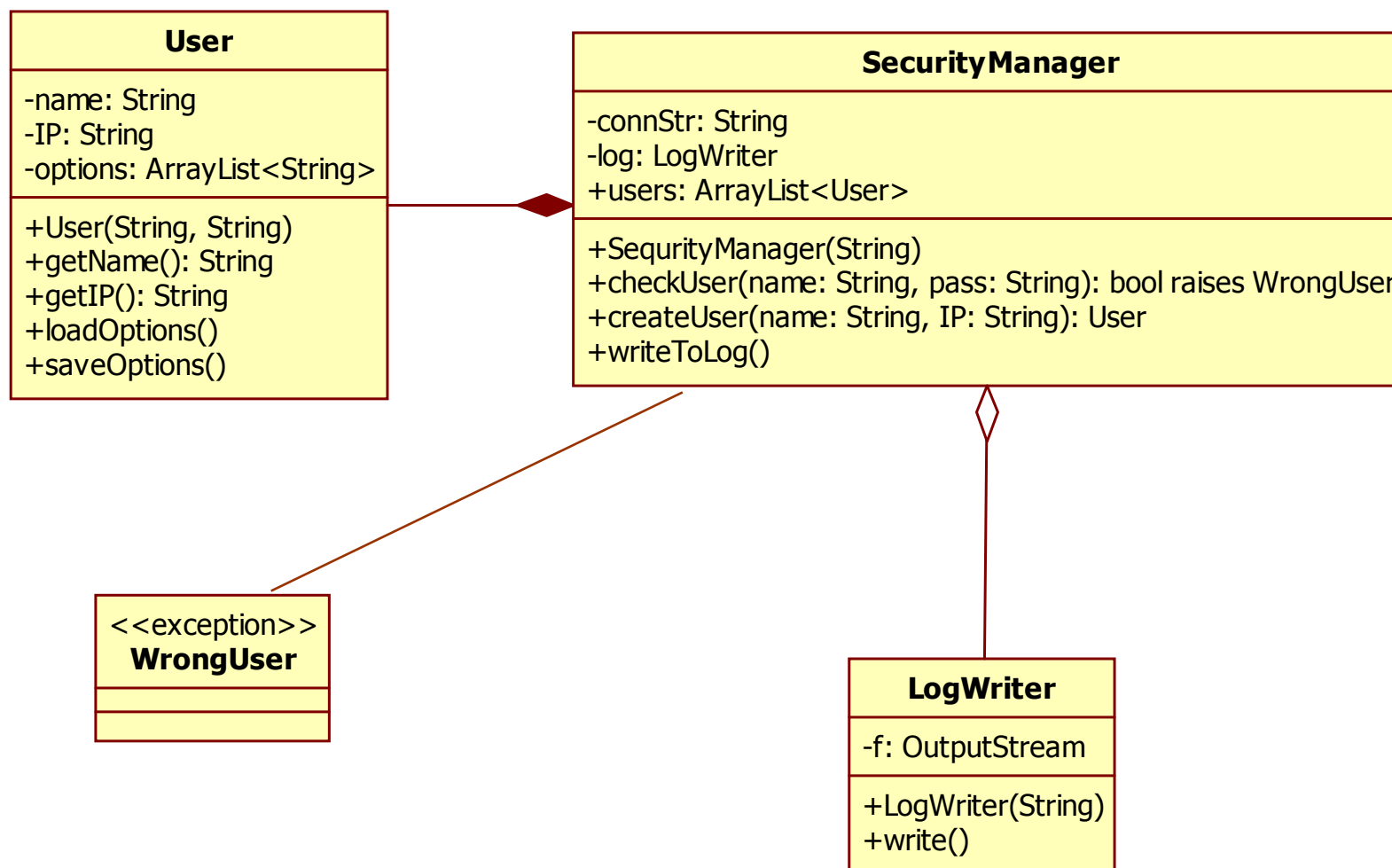


# Взаимосвязи классов

- Ассоциация (Associate)



- Типы ассоциаций : взаимодействие, включение, композиция, агрегация



# C++



```
//  
//  
// Generated by StarUML(tm) C++ Add-In  
//  
// @ Project : Untitled  
// @ File Name : User.h  
// @ Date : 15.03.2012  
// @ Author :  
//  
//
```

# C++

```
#if !defined(_USER_H)
#define _USER_H

class User {
public:
    User(string name, string ip);
    string getName();
    string getIP();
    void loadOptions();
    void saveOptions();
private:
    string name;
    string IP;
    list<string> options;
};

#endif // _USER_H
```





# C++

```
#include "User.h"

User::User(string name, string ip) {

}

string User::getName() {

}

string User::getIP() {

}

void User::loadOptions() {

}

void User::saveOptions() {

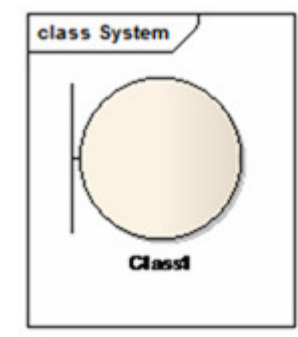
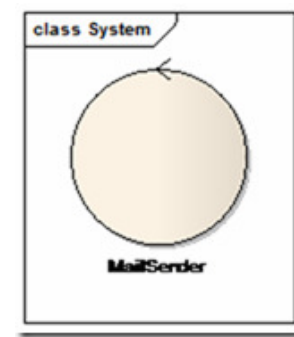
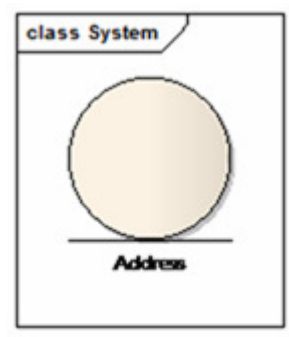
}
```



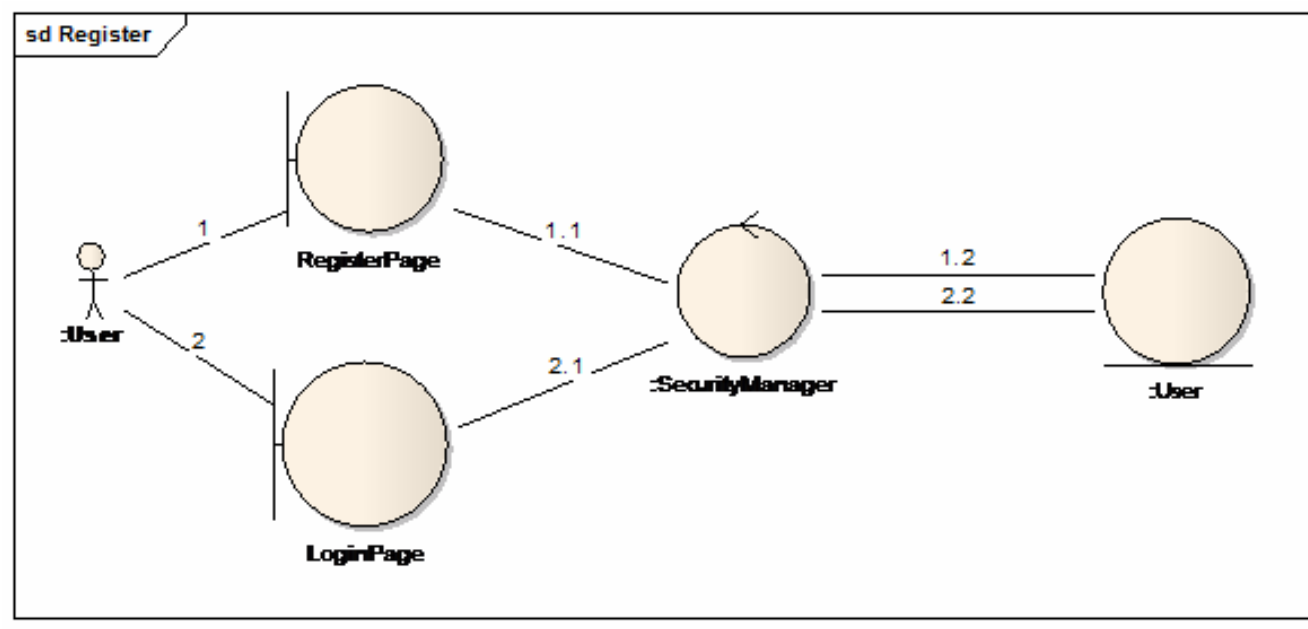
# Диаграммы коммуникаций



- Данный тип диаграмм также относится к поведенческим диаграммам и предназначен для описания коммуникаций между элементами системы
- Для большей наглядности используются альтернативные значки классов – сущность, контроллер, ограничитель, документ и т.д.



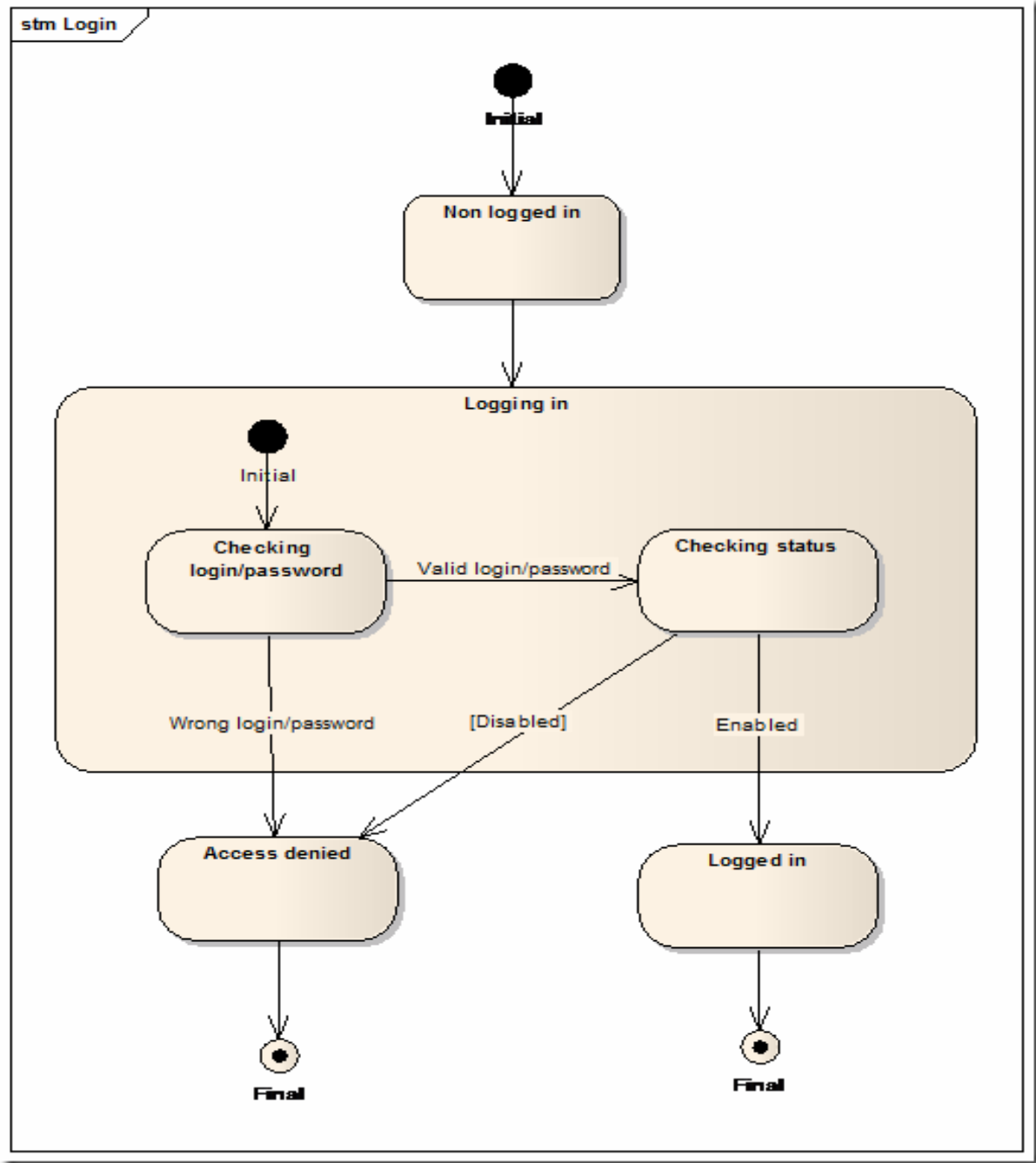
# Диаграммы коммуникаций



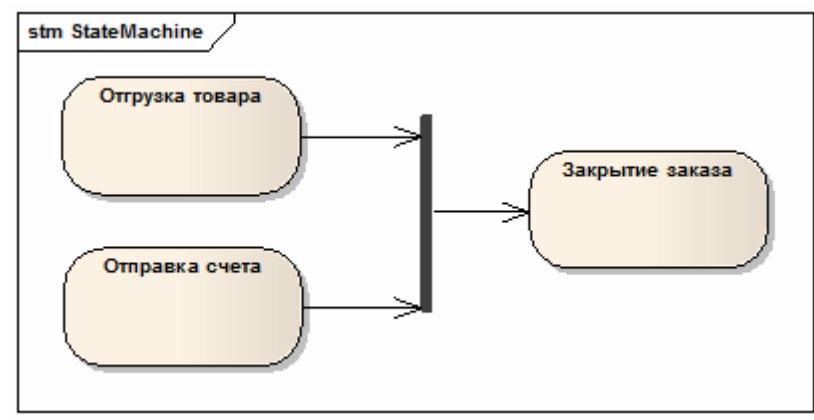
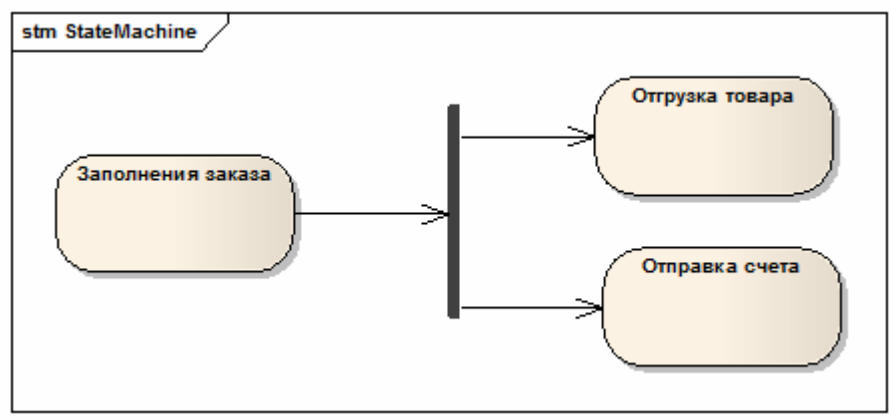
# Диаграммы состояний (State diagram или State Machine)



- Диаграммы состояний обычно применяются для иллюстрации того, как какой либо один элемент (обычно, один класс) переходит между различными своими состояниями.
- Диаграммы состояний также могут применяться как для описания состояний классов, так и бизнес-объектов



# Параллелизм



# Диаграммы деятельности (Activity diagram)



- Диаграммы деятельности предназначены для описания потоков и последовательностей выполнения работ по реализации некоторого варианта использования.
- В отличие от диаграмм состояний, диаграммы деятельности принимают во внимание не состояние некоторого объекта, а потоки деятельности.

