Cohesion/Coupling problem

Additional lecture

The problem

- Develop a program to manage personal information. The information to be manage includes name and email. For name, the program can extract first name and last name, can check if the last name is a popular one in Vietnam, can compare names based on first name and last name. For email, the program can check if it is valid email and extract username and domain accordingly. Emails can also be compared based on username and domain. The program can tell if an email if of .vn domain or not. Lastly, the program can check if the email of a user is of convention lastname.firstname or not. If the user is a Vietnamese, the email will also be checked if of .vn domain or not.

Module 1: Email Processing

book checkFormat(string email) //check the format u@v
string extractUsername(string email)
    // call checkFormat
    // call extractFirstPart
string extractDomain(string email)
    // call check Format first
    // call extractLastPart
bool checkVNdomain(string email)
    //call extractDomain,
    //check if domain ended by .vn
int compareEmailsbyUsername(string email1, string email2)
    //call extractUsername
    //call compareStrings
int compareEmailsbyDomain(string email1, string email2)
    //call extractDomain
    //call compareStrings

Module 3: Util

int compareStrings(string s1, string s2)
string copyString(string s, int from, int to) // string = “ABCBD”, int = 1, to = 3 → “BCB”
int getLength(string s)
int findOccurencefromFirst(string s, char c)
    // string = “ABCBD”, char = ‘B’ → return 1
int findOccurencefromLast(string s, char c) // string = “ABCBD”, char = ‘B’ → return 3
string extractFirstPart (string s, char c)
    // string = “ABCBD”, char = ‘C’ → “AB”, using findOccurencefromFirst and copyString
string extractLastPart (string s, char c)
    // string = “ABCBD”, char = ‘C’ → “AB”, using findOccurencefromLast and copyString

Module 2: Name Processing

string extractFirstName(string name)
    // use extractFirstPart
string extractLastName(string name)
    // use extractLastPart
string compareNames(string name1, string name2)
    // call extractFirstName
    // use CompareStrings to compare
    // if first names are the same
        //call extractLastName
        //use CompareStrings to compare
bool checkPopVNNames(string name)
    // call extractLastName
    // if last name among “Nguyen”, “Tran”, “Le”, “Ly” → return true
string checkEmailConvention(string name, string email)
    // call extractUsername
    // call extractLastName
    // call extractFirstName
    // check username == FirstName.LastName by CompareString
    // call checkVNdomain and call checkPopVNNames for further check

Module 1: Extractor

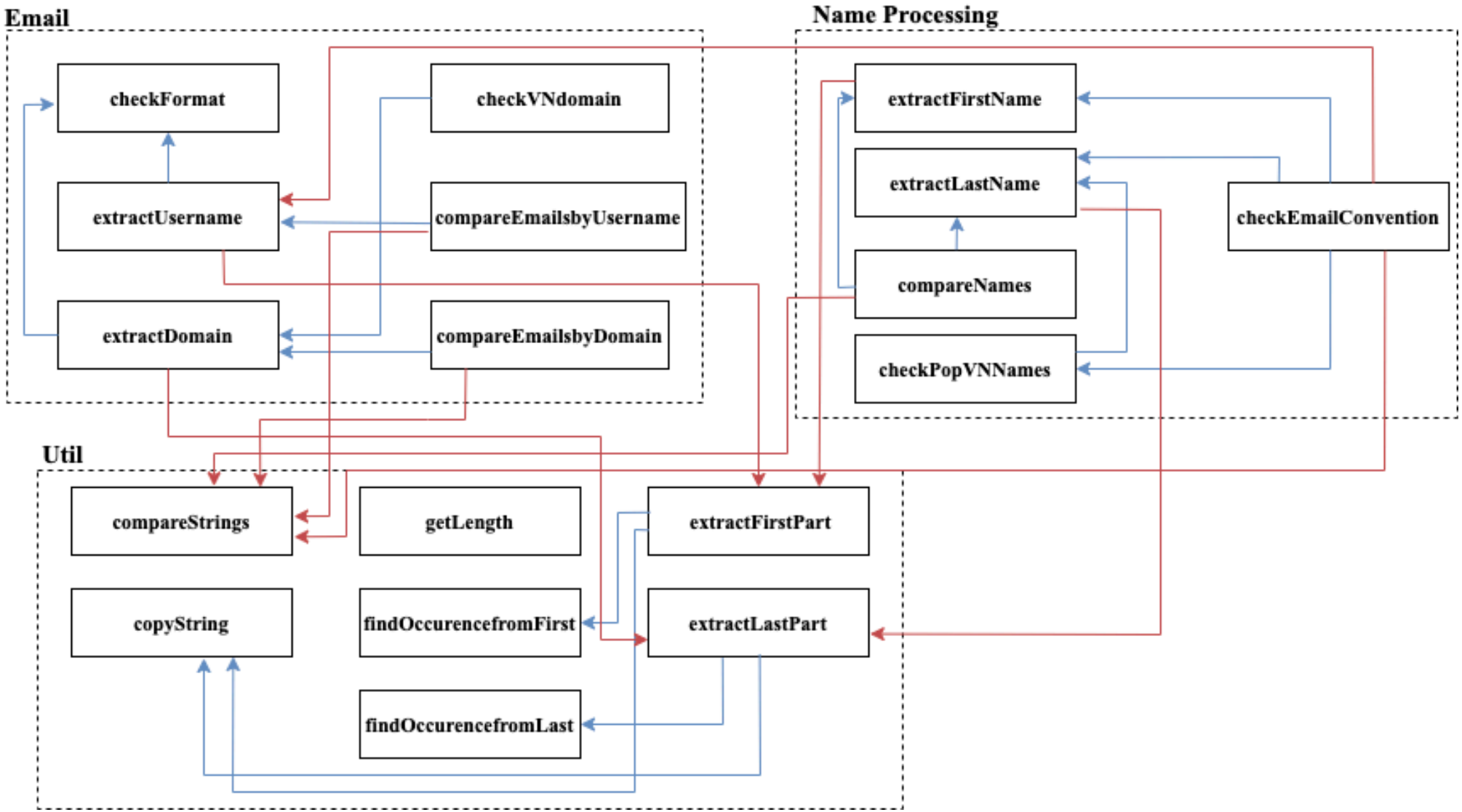
```
string extractUsername(string email)
        // call checkFormat
        // call extractFirstPart
string extractDomain(string email)
        // call check Format first
        // call extractLastPart
string extractFirstPart (string s, char c)
    // string = "ABCBD", char = 'C' → "AB", using findOccurencefromFirst and copyString
string  extractLastPart(string s, char c)
    // string = "ABCBD", char = 'C' → "AB", using findOccurencefromLast and copyString
string extractFirstName(string name)
        // use  extractFirstPart
string extractLastName(string name)
        // use  extractLastPart
```

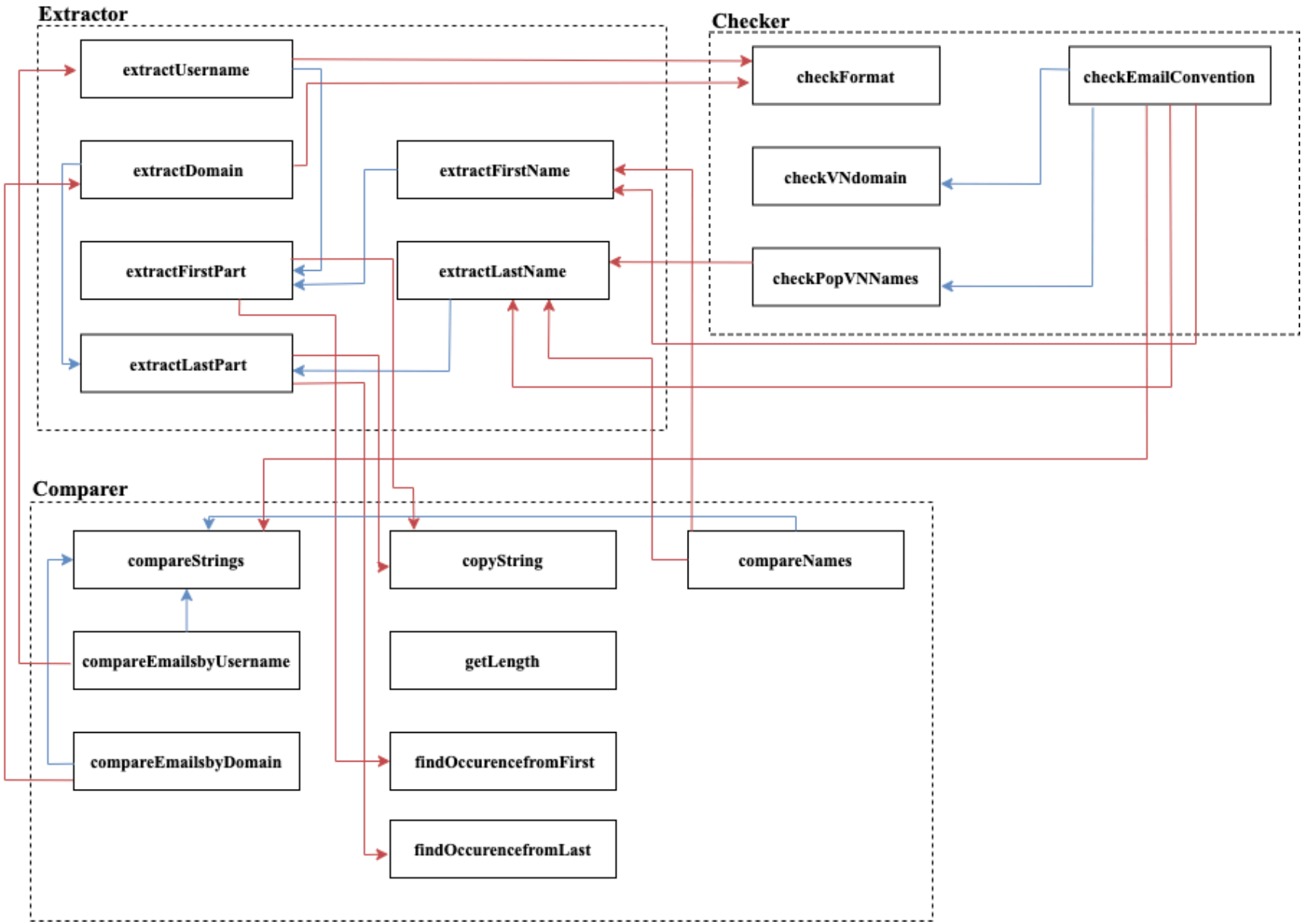
Module 2: Checker

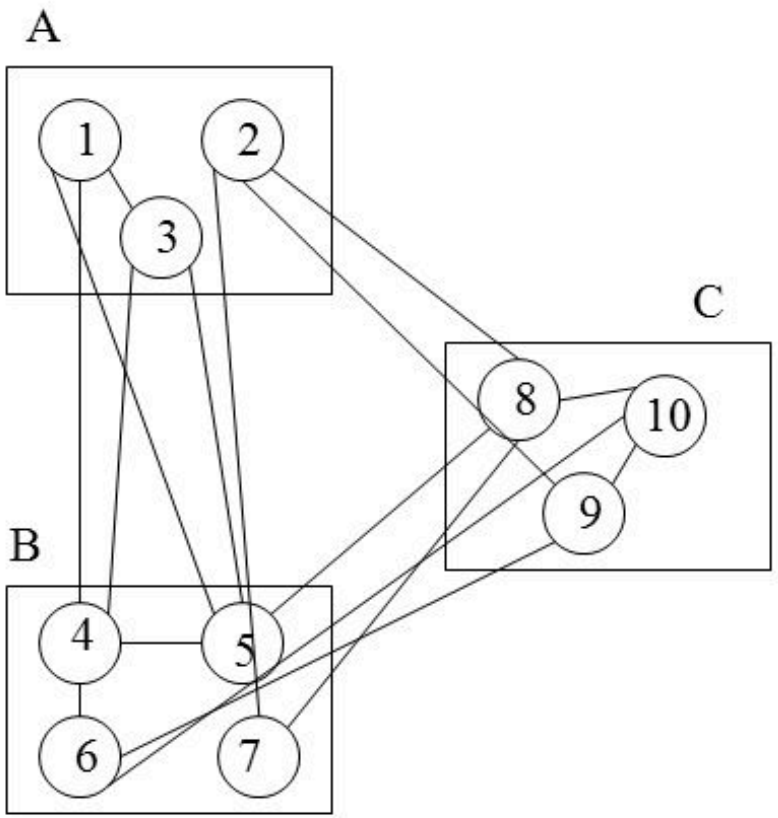
```
book checkFormat(string email) //check the format u@v
bool checkVNdomain(string email)
        //call extractDomain,
        // check if domain ended by .vn
 bool checkPopVNNames(string name)
        // call extractLastName
        // if last name among "Nguyen", "Tran", "Le", "Ly"  → return true
string checkEmailConvention(string name, string email)
        // call extractUsername
        // call extractLastName
        // call extractFirstName
        // check username == FirstName.LastName by CompareString
        // call checkVNdomain and call checkPopVNNames for further check
```

Module 3: Comparer

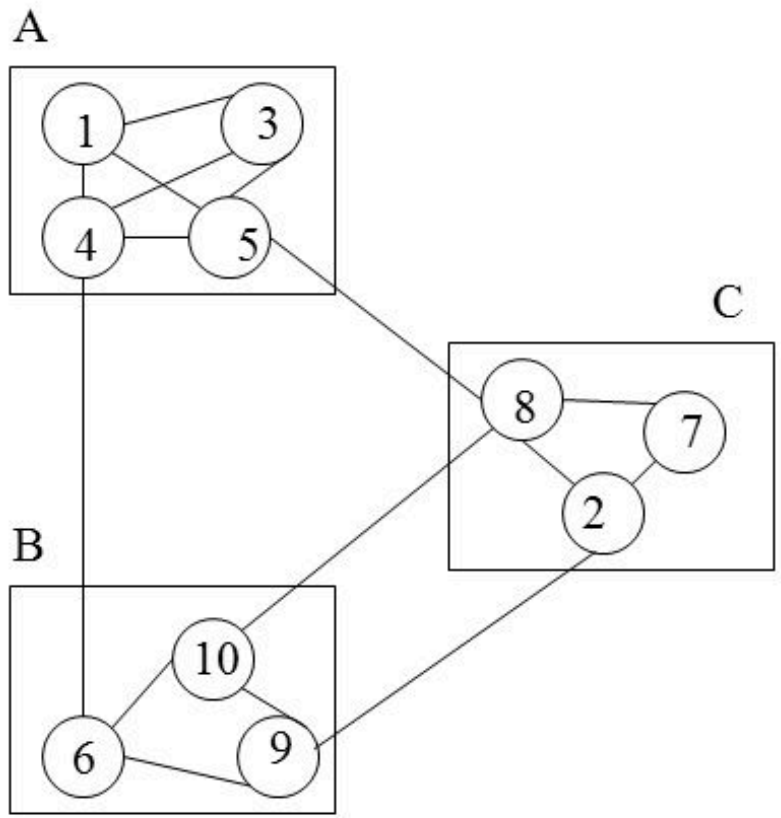
```
int compareStrings(string s1, string s2)
int compareEmailsbyUsername(string email1, string email2)
        //call extractUsername
        //call compareStrings
int compareEmailsbyDomain(string email1, string email2)
        //call extractdomain
        //call compareStrings
string copy(string s, int from, int to) // string = "ABCBD", int = 1, to = 3 → "BCB"
int getLength(string s)
int findOccurencefromFirst(string s, char c) // string = "ABCBD", char = 'B' → return 1
int findOccurencefromLast(string s, char c) // string = "ABCBD", char = 'B' → return 3
string compareNames(string name1, string name2)
        // call extractFirstName
        // use CompareStrings to compare
```


Email
checkFormat
checkVNdomain
extractUsername
compareEmailsbyUsername
extractDomain
compareEmailsbyDomain
Name Processing
extractFirstName
extractLastName
checkEmailConvention
compareNames
checkPopVNNames
Util
compareStrings
getLength
extractFirstPart
copyString
findOccurencefromFirst
extractLastPart
findOccurencefromLast


Extractor
extractUsername
extractDomain
extractFirstPart
extractLastPart
extractFirstName
extractLastName
Checker
checkFormat
checkEmailConvention
checkVNdomain
checkPopVNNames
Comparer
compareStrings
compareEmailsbyUsername
compareEmailsbyDomain
copyString
getLength
findOccurencefromFirst
findOccurencefromLast
compareNames



Bad modularization:
low cohesion, high coupling



Good modularization:
high cohesion, low coupling