



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ I – Năm học 2022-2023

MÃ LƯU TRỮ
(do phòng KT-ĐBCL ghi)
CK22231
CSC13106

Tên học phần: Kiến trúc phần mềm Mã HP: CSC13106
Thời gian làm bài: 120 phút Ngày thi: 30/12/2022
Ghi chú: Sinh viên [☐ được phép / ☒ không được phép] sử dụng tài liệu khi làm bài.

Lưu ý: Sinh viên có thể sử dụng bất kỳ ngôn ngữ lập trình phổ biến nào để trả lời các câu hỏi.

PHẦN 1: ĐỐI TƯỢNG PHÂN TÁN CÓ THỂ LƯU TRỮ (CACHE) MỘT SỐ THUỘC TÍNH CỤ THỂ

Trong quá trình học, anh/chị đã được giới thiệu về đối tượng phân tán (Distributed Object). Ý tưởng về đối tượng phân tán được sử dụng khá phổ biến trong các kiến trúc và ứng dụng phần mềm khác nhau. Trong đề bài này, anh/chị sẽ xây dựng một giải pháp dựa trên ý tưởng đối tượng phân tán, nhưng có bổ sung yêu cầu sau đây: Mỗi thuộc tính dữ liệu của đối tượng (bản gốc, ở server) có thời gian tối đa cho phép các đối tượng tham chiếu (ở các client) có thể lưu trữ lại (cache) giá trị của thuộc tính.

☐ Quy ước: trong phần 1 của đề bài, các class có tên bắt đầu bằng chữ M (Main) là các lớp đối tượng ở phía server, phục vụ việc xử lý trên các đối tượng là bản gốc; các class có tên bắt đầu bằng chữ R (Ref) là các lớp đối tượng ở phía client, phục vụ việc xử lý trên các đối tượng là bản tham chiếu.

Lớp MAttribute (tại server) biểu diễn một thuộc tính cho đối tượng bản gốc ở phía server.

```
class MAttribute {  
    public string Name;    // tên thuộc tính  
    public string Value;    // giá trị thuộc tính  
    public int CachedTime; // thời gian (giây) cho phép cache giá trị thuộc tính  
    // thời gian (giây) mặc định cho phép cache giá trị thuộc tính  
    public static int DefaultCachedTime = 60;  
  
    public MAttribute(string name, string value, int cachedTime) {  
        Name = name;  
        Value = value;  
        CachedTime = cachedTime;  
    }  
}
```

Lớp MObject thể hiện đối tượng bản gốc tại server.

```
class MObject {  
    public int ID; // định danh duy nhất, phân biệt giữa các đối tượng tại server  
    protected Dictionary<string, MAttribute> Attributes  
        = new Dictionary<string, MAttribute>(); // tự điển các thuộc tính, khóa là tên  
  
    public MObject() {...} // constructor cho 1 đối tượng MObject  
    public string this[string sAttributeName] {...} // get và set giá trị 1 thuộc tính  
}
```

Họ tên người ra đề/MSCB:
Họ tên người duyệt đề:

Chữ ký:
Chữ ký:

(Đề thi gồm 6 trang)
[Trang 1/6]



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ I – Năm học 2022-2023

MÃ LƯU TRỮ
(do phòng KT-ĐBCL ghi)

Lớp đối tượng `MObjectManager` quản lý việc tạo lập và đăng ký các đối tượng `MObject`, đồng thời cho phép truy cập (get và set) các thuộc tính của các đối tượng `MObject` tại server.

```
class MObjectManager {  
    // tự điển các đối tượng bản gốc tại server, khóa là ID của đối tượng  
    public static Dictionary<int, MObject> objects = new Dictionary<int, MObject>();  
  
    // đăng ký 1 đối tượng MObject vào tự điển tại server  
    public static void Register(MObject mObject) {...}  
    // tìm theo ID và trả về đối tượng MObject trong tự điển, trả về null nếu không có  
    private static MObject FindObjectByID(int id) {...}  
    // tạo ra đối tượng MObject và trả về ID  
    public static int CreateObject() {...}  
  
    // cập nhật thuộc tính có tên là sAttributeName của đối tượng có định danh là id  
    public static bool SetObjectAttributeValue(  
        int id, string sAttributeName, string newValue) {...}  
  
    // lấy giá trị thuộc tính tên sAttribute trong đối tượng có định danh là id  
    public static string GetObjectAttributeValue(int id, string sAttributeName) {...}  
    // lấy thời gian cho phép cache giá trị của thuộc tính tên sAttributeName trong  
    // đối tượng có định danh là id  
    public static int GetObjectAttributeCachedTime(int id, string strAttributeName){...}
```

Câu 1 (1 điểm): Hãy cài đặt phương thức `this[string sAttributeName]` trong lớp `MObject` để cho phép lấy giá trị và cập nhật giá trị của thuộc tính có tên là `sAttributeName` trong đối tượng của lớp `MObject`.

➤ Hướng dẫn:

- Sử dụng `Attributes.ContainsKey(string)` để kiểm tra xem trong tự điển `Attributes` có khóa (là `string`) cho trước hay không.
- Khi lấy giá trị của thuộc tính có tên `sAttributeName`, cần kiểm tra xem có tồn tại thuộc tính này trong đối tượng hay không. Nếu không tồn tại, trả về chuỗi rỗng (`""`).
- Khi cập nhật giá trị của thuộc tính có tên `sAttributeName`, nếu thuộc tính này đã tồn tại trong đối tượng và giá trị hiện tại khác với giá trị cần cập nhật thì tiến hành cập nhật giá trị thuộc tính. Nếu thuộc tính này chưa tồn tại thì bổ sung vào đối tượng (nên sử dụng constructor của lớp `MAttribute` với thời gian cache mặc định là `MAttribute.DefaultCachedTime`).

Câu 2 (1 điểm): Hãy cài đặt phương thức `FindObjectByID()` của lớp `MObjectManager` cho phép tìm và trả về đối tượng `MObject` với định danh (`id`) cho trước (trả về null nếu không tồn tại). Sau đó, cài đặt phương thức `GetObjectAttributeCachedTime()` của lớp `MObjectManager` để lấy thời gian cho phép cache giá trị một thuộc tính của đối tượng với định danh (`id`) cho trước.

Họ tên người ra đề/MSCB:
Họ tên người duyệt đề:

Chữ ký:
Chữ ký:

(Đề thi gồm 6 trang)
-[Trang 2/6]



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ I – Năm học 2022-2023

MÃ LƯU TRỮ
(do phòng KT-ĐBCL ghi)

Hướng dẫn: Hàm `GetObjectAttributeCachedTime()` trả về 0 nếu không tồn tại đối tượng với định danh (id) cho trước, hoặc đối tượng không có thuộc tính với tên cho trước (`sAttributeName`).

Câu 3 (1 điểm):

Để lưu trữ (cache) giá trị của thuộc tính tại mỗi đối tượng bản sao (tham chiếu) tại client, ta định nghĩa lớp đối tượng `RAttribute` như sau:

```
public class RAttribute {  
    public string Name; // tên thuộc tính  
    private string LastKnownValue; // giá trị gần đây nhất của thuộc tính  
    public int CachedTime; // thời gian (giây) cho phép cache giá trị thuộc tính  
    public DateTime LastUpdate; // thời điểm lấy giá trị mới nhất từ server  
    public static int DefaultCachedTime = 60; // thời gian cache tối đa mặc định  
  
    public RAttribute(string name, string value, int cachedTime) {...} // constructor  
    public string Value { get { ... } set { ... } } // lấy giá trị/cập nhật LastKnownValue  
    public bool NeedToRefresh() {...} // kiểm tra xem có cần lấy giá trị mới từ server
```

Hãy cài đặt hàm `NeedToRefresh()` của lớp `RAttribute` để *kiểm tra* xem ở thời điểm hiện tại *có cần lấy giá trị mới* của thuộc tính từ đối tượng bản gốc `MObject` tại server hay không. Sau đó, cài đặt phương thức `Value` để lấy giá trị và cập nhật giá trị `LastKnownValue`.

Hướng dẫn:

- Có thể sử dụng `DateTime.Now` để lấy thông tin ngày giờ hiện tại. Điều kiện để cần cập nhật lại giá trị thuộc tính là thời gian (tính bằng giây) từ thời điểm hiện tại đến thời điểm lấy giá trị mới nhất từ server (`LastUpdate`) vượt quá `DefaultCachedTime`.
- Cần lưu ý để tự động ghi nhận thời điểm cập nhật giá trị thuộc tính (`LastUpdate`) mỗi khi cập nhật (set) giá trị thuộc tính trong phương thức `Value`.

Họ tên người ra đề/MSCB:
Họ tên người duyệt đề:

Chữ ký:
Chữ ký:

(Đề thi gồm 6 trang)
{Trang 3/6}



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ I – Năm học 2022-2023

MÃ LƯU TRỮ
(đo phỏng KT-ĐBCL ghi)

Đối tượng thuộc lớp RObject là bản sao (bản tham chiếu – reference) tại client của đối tượng bản gốc (MObject) tại server. Khác với ví dụ trong bài giảng của môn học, mỗi đối tượng thuộc lớp RObject có thể lưu trữ lại (cache) giá trị thuộc tính trong một khoảng thời gian nhất định; mỗi thuộc tính có thể có thời gian cho phép cache giá trị khác nhau.

```
class RObject {
    protected int ID; // định danh của đối tượng bản gốc MObject tại server
    protected Dictionary<string, RAttribute> Attributes
        = new Dictionary<string, RAttribute>(); // tự điển các thuộc tính được cache

    // truy cập (get và set) giá trị thuộc tính với tên cho trước
    public string this[string strAttributeName] { get {...} set {...} }
    // bổ sung một thuộc tính mới vào tự điển
    private void AddNewAttribute
        (string strAttributeName, string value, int cachedTime) {...}
}
```

Lớp đối tượng RObjectManager cho phép kết nối và làm việc với server để tạo lập, liên kết, truy cập (get và set) giá trị thuộc tính các đối tượng bản gốc tại server.

```
class RObjectManager {
    public static int CreateObject() {...} // tạo ra đối tượng tại server và trả về ID
    // lấy giá trị của thuộc tính tên strAttributeName của đối tượng có định danh id
    public static string GetObjectAttributeValue(int id, string strAttributeName) {...}
    // lấy thời gian cho phép cache giá trị thuộc tính tên strAttributeName
    // của đối tượng có định danh id
    public static int GetObjectAttributeCachedTime
        (int id, string strAttributeName) {...}
    // cập nhật giá trị thuộc tính tên strAttributeName của đối tượng có định danh id
    public static bool SetObjectAttributeValue
        (int id, string strAttributeName, string newValue) {...}
}
```

Câu 4 (1 điểm): Hãy cài đặt thao tác *cập nhật giá trị thuộc tính* với tên cho trước trong đối tượng RObject.

➤ **Hướng dẫn:** anh/chị cài đặt trong phần set{...} của phương thức this[string strAttributeName]

- Nếu tự điển Attributes đã có thuộc tính với tên là strAttributeName và giá trị (Value) hiện tại của thuộc tính này khác với giá trị cần cập nhật thì gán giá trị mới cho thuộc tính Value của đối tượng RAttribute tương ứng.
- Nếu tự điển Attributes chưa có thuộc tính với tên là strAttributeName thì bổ sung thuộc tính mới vào tự điển (sử dụng hàm AddNewAttribute(), không cần cài đặt).
- Cần lưu ý là trong cả hai tình huống trên, mỗi khi cập nhật giá trị thuộc tính của bản sao tham chiếu (ở client) phải gọi phương thức RObjectManager.SetObjectAttributeValue() để cập nhật giá trị thuộc tính của bản gốc tại server.

Họ tên người ra đề/MSCB:
Họ tên người duyệt đề:

Chữ ký:.
Chữ ký:.

(Đề thi gồm 6 trang)
[Trang 4/6]



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ I – Năm học 2022-2023

MÃ LƯU TRỮ
(do phòng KT-ĐBCL ghi)

Câu 5 (1 điểm):

Hãy cài đặt thao tác *lấy giá trị cho 1 thuộc tính* với tên cho trước trong đối tượng `RObject`.

➤ **Hướng dẫn:** anh/chị cài đặt trong phần `get{...}` của phương thức `this[string strAttributeName]`

- Nếu tự điển `Attributes` đã có thuộc tính với tên là `strAttributeName` và thuộc tính này cần lấy giá trị mới từ đối tượng bản gốc `MObject` tại server (sử dụng phương thức `NeedToRefresh()` của `Attributes[strAttributeName]`) thì:
 - Gọi phương thức `RObjectManager.GetObjectAttributeValue()` để lấy giá trị thuộc tính có tên `strAttributeName` của đối tượng có định danh là `ID` tại server.
 - Cập nhật giá trị (`Value`) của thuộc tính `Attributes[strAttributeName]` theo giá trị mới nhận được từ server.
- Nếu tự điển `Attributes` chưa có thuộc tính với tên là `strAttributeName` thì:
 - Gọi phương thức `RObjectManager.GetObjectAttributeValue()` để lấy giá trị thuộc tính có tên `strAttributeName` của đối tượng có định danh là `ID` tại server.
 - Gọi phương thức `RObjectManager.GetObjectAttributeCachedTime()` để lấy thời gian (giây) cho phép cache giá trị thuộc tính có tên `strAttributeName` của đối tượng có định danh là `ID` tại server.
 - Sử dụng phương thức `AddNewAttribute()` để bổ sung thuộc tính `RAttribute` với thông tin có được vào tự điển `Attributes`.
- Trả về giá trị (`Value`) của đối tượng thuộc tính `Attributes[strAttributeName]`.

Câu 6 (1 điểm): Hãy *đề xuất giải pháp và cài đặt bổ sung* các hàm/thuộc tính cần thiết để `RObjectManager` có thể quản lý được tất cả các đối tượng `RObject` được tạo ra tại mỗi client, đồng thời cho phép xóa toàn bộ giá trị cache của tất cả các đối tượng `RObject` tại client này.

➤ **Hướng dẫn:** anh/chị tham khảo giải pháp quản lý toàn bộ các đối tượng `MObject` tại server để đề xuất giải pháp quản lý toàn bộ đối tượng `RObject` tại client. Lưu ý là có thể có nhiều đối tượng `RObject` có cùng ID (tức là tham chiếu cùng đến 1 đối tượng bản gốc `MObject` tại server).

Câu 7 (1 điểm): Hãy *đề xuất giải pháp và cài đặt minh họa* để cho phép khi cập nhật giá trị đối tượng thuộc tính `MAttribute` tại server thì tất cả các đối tượng bản sao tham chiếu (`RObject`) tại tất cả client cũng được thông báo để xóa giá trị cache (nếu có) và cập nhật giá trị mới.

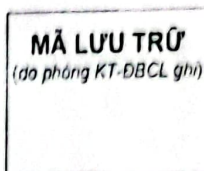
Họ tên người ra đề/MSCB:
Họ tên người duyệt đề:

Chữ ký:
Chữ ký:

(Đề thi gồm 6 trang)
[Trang 5/6]



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ I – Năm học 2022-2023



Câu 8 (1 điểm): Kiến trúc để quản lý đối tượng thông tin được trình bày trong đề bài này chỉ là giải pháp được đơn giản hóa nhằm minh họa việc vận dụng kiến thức đã học vào một tình huống có bổ sung thêm yêu cầu mới.

Việc cache lại giá trị một số thuộc tính cho phép ứng dụng ở client vẫn có thể hoạt động trong một thời gian cho phép mà không cần phải truy vấn liên tục thông tin từ server. Tuy nhiên, tùy vào ứng dụng và nghiệp vụ cụ thể, mỗi thuộc tính có thời gian cho phép cache khác nhau.

Một ví dụ cho việc sử dụng kiến trúc này là hệ thống quản lý thông tin xe buýt. Khi đó, thông tin về các trạm xe buýt cũng như lộ trình các tuyến có thể được lưu (cache) lại tại mỗi client (là thiết bị di động) trong khoảng thời gian 1 tháng. Sau đó, thông tin này có thể được cập nhật lại từ thông tin bản gốc tại server.

Anh/chị hãy *trình bày một ví dụ khác* cho việc ứng dụng kiến trúc để quản lý thông tin đối tượng có khả năng cache giá trị thuộc tính.

PHẦN 2 – AWS ACADEMY

Câu 9 (1 điểm)

Hãy trình bày một vấn đề mà anh/chị đã học và thấy tâm đắc trong quá trình tìm hiểu thông qua chương trình AWS Academy.

PHẦN 3 – TRẢI NGHIỆM THỰC TẾ

Câu 10 (1+ điểm)

Hãy trình bày một ví dụ cụ thể về việc ứng dụng kiến trúc phần mềm trong đồ án hay đề tài mà anh/chị *đã thực hiện* hoặc *đã tìm hiểu*. Cần trình bày rõ: (a) tình huống, yêu cầu thực tế; (b) giải pháp đề xuất; (c) ưu điểm và hạn chế của giải pháp.

-HẾT-

Họ tên người ra đề/MSCB:
Họ tên người duyệt đề:

Chữ ký:
Chữ ký:

(Đề thi gồm 6 trang)
[Trang 6/6]