
WeSketch Team

Stevica Stojković, 15426

Stefan Stanković, 15382

WeSketch

Online alat za kolaborativno vektorsko crtanje

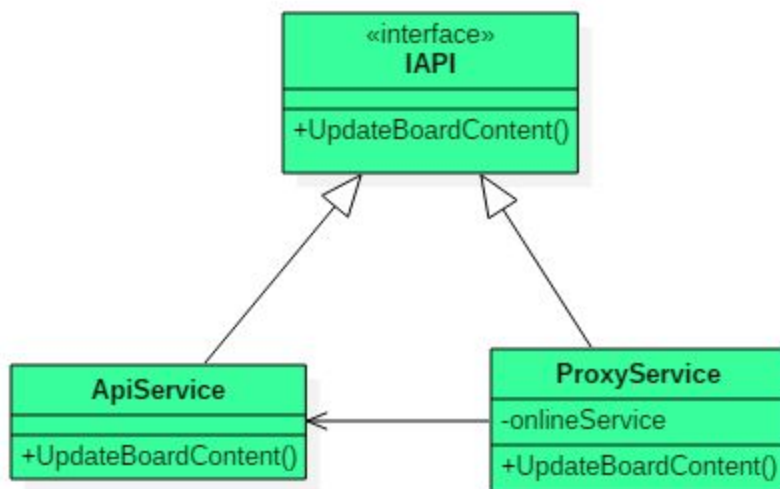
Sadržaj

Sadržaj	0
Obrasci korišćeni tokom izrade projekta	1
Proxy obrazac	1
Template method obrazac	2
Podrška za više načina sinhronizacije	4
Factory Method	4
Strategy	5
Unit of Work i Repository	6

Obrasci korišćeni tokom izrade projekta

Proxy obrazac

Proxy obrazac je upotrebljen sa ciljem da od korisnika sakrije prisustvo ili odsustvo konekcije sa serverom. Korisniku su na raspolaganju različite opcije u aplikaciji u zavisnosti od konekcije.



```

public void EnterQueue(User user, Board board)
{
    if(!hasInternetConnection)
        return;

    onlineService.EnterQueue(user, board);
}

public Board GetBoardById(int id)
{
    if (hasInternetConnection)
        return onlineService.GetBoardById(id);

    Board board = Global.CurrentUser.Boards.Find(x => x.Id == id);
    return board != null ? board : new Board();
}

public List<User> GetBoardCollaborators(Board board)
{
    if (hasInternetConnection)
        return onlineService.GetBoardCollaborators(board);

    return board?.Collaborators.Collaborators;
}

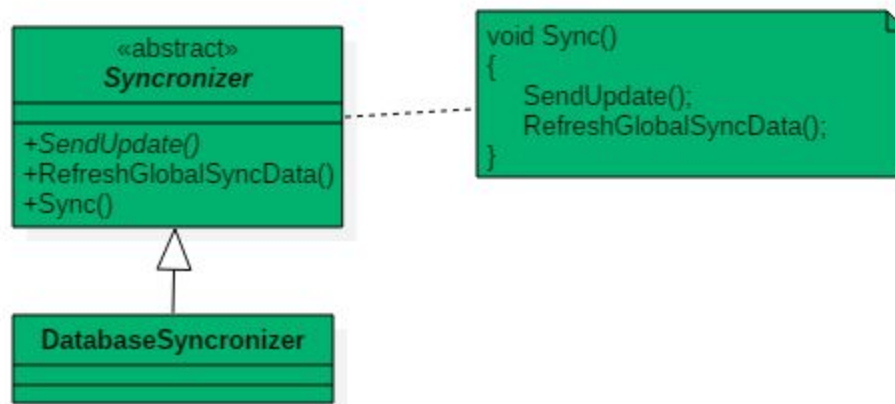
public List<Board> GetMyBoards(User user)
{
    if (hasInternetConnection)
        return onlineService.GetMyBoards(user);

    var boards = user.Boards;
    return boards;
}

```

Template method obrazac

Kada se ponovo uspostavi konekcija sa serverom, potrebno je sinhronizovati promene u offline mode-u sa stanjem u bazi podataka. Kako bi se definisali koraci sinhronizacije, upotrebljen je Template method.



```

public abstract class Synchronizer
{
    public abstract void SendUpdate(SyncerData data);

    public void RefreshGlobalSyncData()
    {
        Global.Syncer.BoardsToCreate = new List<CommonBoard>();
        Global.Syncer.BoardsToUpdate = new List<CommonBoard>();
        Global.Syncer.BoardsToDelete = new List<CommonBoard>();
    }

    public void Sync()
    {
        SyncerData data = Global.Syncer;
        SendUpdate(data);
        RefreshGlobalSyncData();
    }
}
  
```

```

public class DatabaseSynchronizer : Synchronizer
{
    private string ServerURI = Global.ServerURI;
    private HubConnection connection;
    IHubProxy boardHub;

    public DatabaseSynchronizer()
    {
        connection = new HubConnection(ServerURI);
        connection.TraceLevel = TraceLevels.All;
        ServicePointManager.DefaultConnectionLimit = 10;
        boardHub = connection.CreateHubProxy("BoardHub");
        connection.Start().Wait();
    }

    public override void SendUpdate(SyncerData data)
    {
        Application.Current.Dispatcher.Invoke(DispatcherPriority.Normal, new Action(() =>
        {
            boardHub.Invoke("SyncOfflineModeChanges", data);
        }));
    }
}

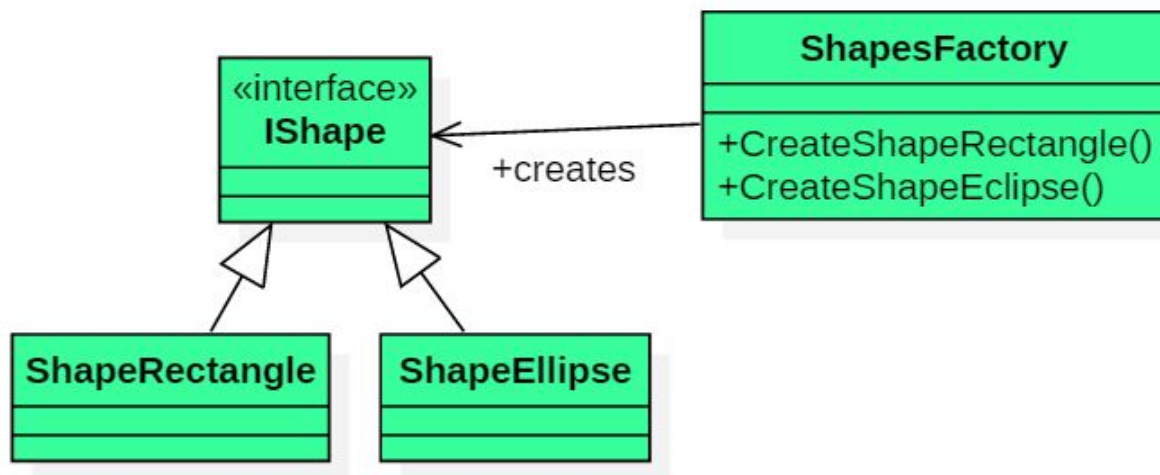
```

Podrška za više načina sinhronizacije

Moguće je lako dodavanje novih načina sinhronizacije(na primer upis u datoteku na klijentu) izvođenjem iz klase Synchronizer i predefinisanjem odgovarajuće metode. Na taj način je moguće proširenje primenom Strategy obrasca i odluka o načinu sinhronizacije u runtime-u.

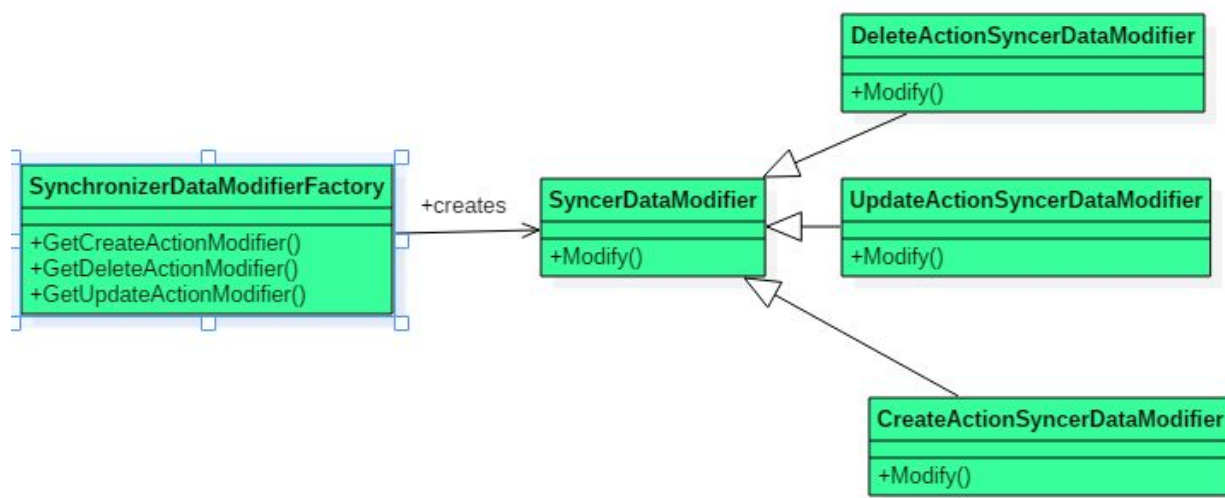
Factory Method

Factory Method je iskorišćen za instanciranje odgovarajućih objekata iz hijerarhije klasa(odgovarajućih shape-ova).



Strategy

U offline mode-u, postoje različite liste za table(crteže) koje treba obrisati, kreirati ili modifikovati. U zavisnosti od toga da li je tabla postojala pre nestanka konekcije ili je kreirana nakon toga, potrebno je izvršiti različite akcije nad listama kada se neka tabla briše, kreira ili ažurira. Problem je rešen primenom Strategy obrasca.

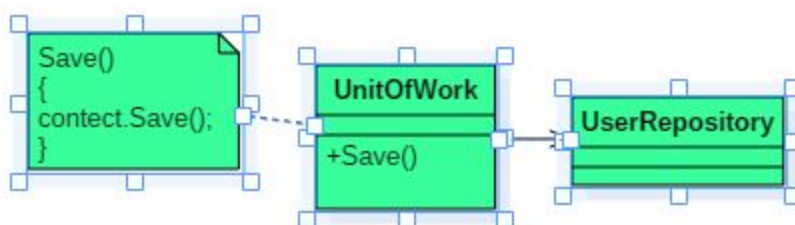


Konkretna klasa iz hijerarhije Strategy-a se dobija upotrebom Factory Method obrasca.

Unit of Work i Repository

Repository pattern je upotrebljen sa ciljem da svaka Repository klasa enkapsulira CRUD operacije za određeni entitet u bazi podataka.

Unit of Work enkapsulira Repository klase. Ponaša se kao kontroler obraćanja bazi podataka i vodi računa učitanim objektima. Ova primena je postignuta enkapsuliranjem context Entity Framework-a u okviru Unit of Work-a. Svi učitani objekti u jednoj transakciji su vezani za taj context.



```

public void SyncUpdatedBoards(List<CommonBoard> boards)
{
    foreach (var boardToUpdate in boards)
    {
        Board board = unitOfWork.BoardRepository.GetById(boardToUpdate.BoardId);
        board.Content = boardToUpdate.Content;
        board.Description = boardToUpdate.Description;
        board.Title = boardToUpdate.Title;
        board.ActiveBoard = true;
        unitOfWork.BoardRepository.Update(board);
    }
    unitOfWork.Save();
}

```

Observer

S obzirom da aplikacija radi u online i offline režimu, nekim komponentama je neophodno to da znaju kada korisnik izgubi ili povрати internet konekciju. Te komponente implementiraju interfejs `IConnectionObserver` (observer). Komponenta zadužena za obaveštavanje o promeni konektivnosti je `ConnectivityNotifier` (subject)

