



ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РЕСПУБЛИКИ КАЗАХСТАН
MINISTRY OF EDUCATION AND SCIENCE
OF THE REPUBLIC OF KAZAKHSTAN



Л. Н. ГУМИЛЕВ АТЫНДАҒЫ
ЕУРАЗИЯ ҰЛТТЫҚ УНИВЕРСИТЕТІ
ЕВРАЗИЙСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ ИМ. Л. Н. ГУМИЛЕВА
GUMILYOV EURASIAN
NATIONAL UNIVERSITY



Студенттер мен жас ғалымдардың
«Ғылым және білім - 2015»
атты X Халықаралық ғылыми конференциясының
БАЯНДАМАЛАР ЖИНАҒЫ

СБОРНИК МАТЕРИАЛОВ
X Международной научной конференции
студентов и молодых ученых
«Наука и образование - 2015»

PROCEEDINGS
of the X International Scientific Conference
for students and young scholars
«Science and education - 2015»

УДК 001:37.0
ББК72+74.04
Ғ 96

Ғ96

«Ғылым және білім – 2015» атты студенттер мен жас ғалымдардың X Халық. ғыл. конф. = X Межд. науч. конф. студентов и молодых ученых «Наука и образование - 2015» = The X International Scientific Conference for students and young scholars «Science and education - 2015». – Астана: <http://www.eni.kz/ru/nauka/nauka-i-obrazovanie-2015/>, 2015. – 7419 стр. қазақша, орысша, ағылшынша.

ISBN 978-9965-31-695-1

Жинаққа студенттердің, магистранттардың, докторанттардың және жас ғалымдардың жаратылыстану-техникалық және гуманитарлық ғылымдардың өзекті мәселелері бойынша баяндамалары енгізілген.

The proceedings are the papers of students, undergraduates, doctoral students and young researchers on topical issues of natural and technical sciences and humanities.

В сборник вошли доклады студентов, магистрантов, докторантов и молодых ученых по актуальным вопросам естественно-технических и гуманитарных наук.

УДК 001:37.0
ББК 72+74.04

ISBN 978-9965-31-695-1

©Л.Н. Гумилев атындағы Еуразия
ұлттық университеті, 2015

обеспечения виртуальной инфраструктуры (включая ПО, установленное на сами ВМ), сетевой контроль и защиту гипервизора от сетевых атак, а также управление конфигурациями. Решение гарантирует соблюдение заданных с точки зрения ИБ параметров и позволяет управлять изменениями и уязвимостями.

Список использованных источников

1. Toby Velte; Anthony Velte; Robert Elsenpeter; Robert C. Elsenpeter: Cloud computing, 2009
2. <http://www.computerra.ru/95632/oblachnaya-zashhita-tehnologiya-protiv-ideologii/>
3. <http://www.osp.ru/lan/2013/04/13035155/>

ОӘЖ 004.27

ЖОБАНЫҢ ҮШ ДЕҢГЕЙЛІ АРХИТЕКТУРАСЫ

Каракулов Ерлан Махсұтович

Л.Н. Гумилев атындағы ЕҰУ, 6М070400 – “Есептеу техникасы және бағдарламалық қамтамасыз ету” мамандығының магистранты
Ғылыми жетекшісі – К.Т. Искаков

Кез келген жобада қандай паттернды қолданса да, кодтың үш деңгейлі архитектура қағидалары сақталуы қажет [1-3]:

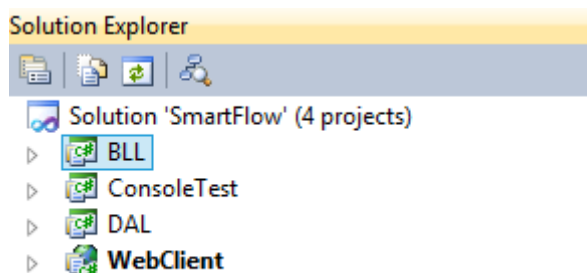
Деңгей DAL (Data Access Layer) – ДҚ-мен жұмысқа жауап береді. Яғни, ешқандай есептеулер жүргізбейді. Оның есебі мәліметтермен тек CRUD амалдарын орындаумен шектеледі.(Create, Read, Update, Delete). Сонымен қатар, транзакциялар орындалуы, ДҚ-на қосылу, sql функцияларын шақыру, процедуралар орындау, яғни, көріп тұрғандарыңыздай, бұл деңгейді біз тікелей СУБД-мен жұмыс жасаймыз.

Деңгей BLL (Business Logic Layer) – жоба құру кезінде қажет болатын барлық есептеулерді қамтиды. Мысалы, менің жобамда тауарды белгілі бір сомаға сату амалы бар (жеңілдіктермен бірге болуы мүмкін), сонда бір транзакцияда ол тауардың сатылуы, қоймадан тауар санының азаюы, компания табысының артуы, т.с.с. амалдар орындалуы тиіс. Байқағаныңыздай, BLL өз ішінде DAL-ды қолданады, керісінше емес.

Деңгей UI (User Interface) – серверден келген мәліметтерді көрсетумен ғана айналысады. Бізде Application server, Database Server бар. Қосымшалар сервері қолданушының жіберген сұранымдарын қабылдап, талдау жасайды. Кейін, сұранысқа сәйкес HTML кодты клиентке жібереді, өз кезегінде Деректер серверіне сұраным жасауы мүмкін. Клиент тек UI көреді, сонымен байланысады, ал, BLL өз кезегінде DAL-мен байланысады. Деңгейлердің байланысу сызбасы мына түрде болушы еді:

Client <->UI<->BLL<->DAL

Осының бәрін Visual Studio бағдарламалау ортасында көрсетейік:



Сурет 1. Жоба құрылымы

Жоба 4 бөліктен тұрады (Сурет 1):

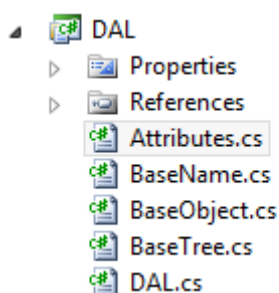
1. BLL – Class Library
2. Console Test – Console Application
3. DAL – Class Library
4. WebClient – Web Framework

Консольдық қосымшада BLL, DAL функцияларын тестілеу үшін қолданамыз, себебі, консольдық қосымшада оларды шақыру, нәтижелерін көру оңайырақ.

ORM (Object Relational Mapping) жайлы айтатын кез келген секілді, себебі осы деңгейде өзім жасаған ORM-технологияны қолданамыз.

ORM – қарапайым сөздермен айтқанда, бұл ДҚ объектіге бағытталған бағдарламалау концепуиясымен байланыстыратын технология. Ол "виртуалды объектілі деректер қорын" құрады. Қазіргі таңда кез келген тілде дайын ORM технологиялар бар. Олардың көмегімен SQL тілін білмей ақ деректермен жұмыс істеу беруге болады. Класстармен жұмыс істей алып, ОББ қағидаларын білсе жеткілікті, толыққанды бакенд жасауға болады.

Бір ескеретін жайт - ORM технология универсалды, кез келген ДҚ-мен жұмыс жасауға арналғандықтан, ресурсты көп талап етеді. Ол технологияны кішкене жобаларда (мілеметтер аз болғандықтан ресурстың көлемі аса байқалмайды), немесе жобаға аз уақыт бөлінгенде қолдануға болады. Менің жағдайымда жоба жасауға уақыт жеткілікті, жоба да аса кішкене емес, сондықтан, өз ORM технологияны құрдық. Жасалған ORM толық емес, ол тек берілген жобаның ғана қажеттіліктерін қанағаттандырады.



Сурет 2. DAL құрылымы

Attributes.cs – атрибуттар класы

BaseObject.cs –түпкі класс

DAL.cs – ДҚ-мен жұмыс жасауға арналған барлық әдістер класы

*Листинг 1.*Attributes.cs құрылымы

```
namespace DAL1
```

```
{  
    [AttributeUsage(AttributeTargets.Property)]  
    public class NotColumnAttribute : Attribute  
    {  
    }  
}
```

Әзірге, біз қосалқы өрістер үшін көрсетілген бір ғана атрибут, үстел бағандар емес болып табылады.

Листинг 2. Түпкі класстың құрылымы

```
namespace DAL1
```

```
{  
    public class BaseObject  
    {  
        public int Id { get; set; }  
    }  
}
```

```

public bool Save()
{
    if (this.Id > 0)
    {
        return DAL.UpdateObject(this);
    }
    else
    {
        return DAL.CreateObject(this);
    }
}

```

Мұнда бір қасиет ID және әдіс Save(). Save – екі әрекетті: сақтау мен өзгертуді орындайды. DAL.cs – толық көрсете алмаймыз. Онда код көп, тек маңызды бөліктеріне тоқталайық.

Листинг 3. Класстың приват бөліктері

```

private static string connString = @"Data Source=localhost;Initial
Catalog=SMART_FLOW_DB;Integrated Security=True";

```

```

private static SqlConnection conn = null;

```

connString – қосылу жолы. Біздің жағдайымызда Integrated Security қолданылады, басқаша, логин мен кілтті сөзді көрсете керек болады.

Conn - мұнда ағымдағы қосылым сақталады.

DAL.cs – өте маңызды және негізгі класс. Бұл деректерді кіру үшін барлық қажетті әдістері. Сыртқы сипаттамасы көптеген әдістерін мақсатын түсіну мүмкін барлық әдістері туралы сипаттаңыз, артық болмас еді.

Листинг 4.

```

public static bool CreateObject(BaseObject obj)
{
    var cmd = CreateCommand();
    var properties = obj.GetType().GetProperties();
    var typeMap = GetTypeMap();
    string columns = string.Empty;
    string query = string.Empty;
    foreach (var pi in properties)
    {
        if (pi.Name != "Id" && !IsInDB(pi))
        {
            object value = pi.GetValue(obj, null);
            if (value != null)
            {
                var dbtype = typeMap[pi.PropertyType];
                var name = string.Format("@{0}", pi.Name.ToLower());
                AddParametersToCommand(ref cmd, name, dbtype, value);
                query += query == string.Empty ? name : "," + name;
                columns += string.IsNullOrEmpty(columns) ? pi.Name : "," + pi.Name;
            }
        }
    }
}

```

```

}
cmd.CommandText= string.Format("insert into {0}({2}) values({1})", obj.GetType().Name, query,
columns);
returnExecuteCommand(cmd);
}

```

Листинг 5.

```

public static boolUpdateObject(BaseObjectobj)
{
varcmd = CreateCommand();
var properties = obj.GetType().GetProperties();
string query = string.Empty;
vartypeMap = GetTypeMap();
foreach (var pi in properties)
{
if (pi.Name != "Id" &&IsInDB(pi))
{
object value = pi.GetValue(obj, null);
if (value != null)
{
vardbtype = typeMap[pi.PropertyType];
var name = string.Format("@{0}", pi.Name.ToLower());
AddParametersToCommand(ref cmd, name, dbtype, value);
query += query == string.Empty ? pi.Name + "=" + name : "," + pi.Name + "=" + name;
}
}
}
cmd.CommandText = string.Format("update {0} set {1} where Id={2} ", obj.GetType().Name,
query, obj.Id);
returnExecuteCommand(cmd);
}

```

Листинг 6.

```

publicstatic T[] GetObjectsByCriteria<T>(SqlParameter[] parameters) where T : BaseObject
{
varcmd = CreateCommand();
string query = GetQuery(typeof(T));
string condition = string.Empty;
foreach (var par in parameters)
{
string col = string.Empty;
for (inti = 1; i<par.ParameterName.Length; i++)
col += par.ParameterName[i];
if (string.IsNullOrEmpty(condition))
{
condition += col + "=" + par.ParameterName;
}
else
{
condition += " and " + col + "=" + par.ParameterName;
}
}
cmd.Parameters.Add(par);

```

```

    }
    cmd.CommandText = query + " WHERE " + condition;
    if (conn.State != ConnectionState.Open) conn.Open();
    var reader = cmd.ExecuteReader(CommandBehavior.CloseConnection);
    returnReaderToArray<T>(reader);
}

```

Листинг 7.

```

public static bool DeleteObject(BaseObject obj)
{
    if (obj.Id > 0)
    {
        var cmd = CreateCommand();
        cmd.CommandText = string.Format("delete from {0} where Id={1}", obj.GetType().Name, obj.Id);
        return ExecuteCommand(cmd);
    }
    return false;
}

```

Қолданылған әдебиет

1. Р. Мартин «Clean Code: A Handbook of Agile Software Craftsmanship», Питер, 2010.
2. С. Макконнел «Code Complete», Питер, 2007.
3. Э.Фримен, «Паттерны проектирования», Питер, 2011.

ӨОЖ 004

DJANGO ФРЕЙМВОРГЫНДАҒЫ WEB-ТЕХНОЛОГИЯЛАРДЫ ЗЕРТТЕУ

Кенебаев Болат Тұрғамбекұлы

bolat_kenabayev@bk.ru

Л.Н. Гумилев атындағы ЕҰУ, 5B070400 – “Есептеу техникасы және бағдарламалық қамтамасыз ету” мамандығының студенті, Астана, Қазақстан
Ғылыми жетекшісі – Ж.О. Оралбекова

Жұмыстың өзектілігі. Web-сайттарды жоспарлау және жүзеге асыру әрқашан көп шығын кетіретін қызметтер қатарына жататыны айқын. Django – тез, өнімділігі жоғары, толыққанды ақпараттық сайттарды жасауда бүгінгі күнде ең үздік фреймворктардың бірі болып табылады. Django көмегімен кез келген ауқымды және оңай кеңейтілетін күрделі қосымшаларды құруға болады. Фреймворк Python тілі негізінде жұмыс жасайды. Web-құраушылар Django көмегімен динамикалық web сайттарды деректер қорын пайдалана отырып, өте қысқа уақытта жасап шығады. Негізгі артықшылықтарының бірі болып django-да жасалған жобалардың кодтардың жылдам өзгертілуінде, соның арқасында қазіргі таңдағы веб пайдаланушылар мен веб құраушылардың жұмыстары едәуір жеңілдетіледі. Мысалы, веб құраушыға контроллер мен сайттың администраторлық бөлігін құрудың қажеті жоқ, себебі Django-да кез келген сайтқа қосуға болатын, бір серверде бірнеше сайтты өзгертетін Django жасалған кодтарды басқару бөлуге болады. Администратор бөлігі кез келген объектіні құруға, өзгертуге және өшіруге мүмкіндік береді және де қолданушыға ыңғайлы интерфейс ұсынады [1]. Django веб-фреймворкін Instagram, Disqus, Mozilla, The Washington Times, Pinterest және т.б. көптеген үлкен сайттар қолданады. Және де Django әр түрлі жобаларда веб компонент ретінде қолданыс тапқан, атап айтсақ, Graphite– график және мониторинг жасау жүйесінде, Яндекс Погода – ауа-райы болжамында және т.б. Жоғарыда айтылғандардың бәрі жұмыстың өзектілігін құрайды [2].

Жұмыстың мақсаты: веб сайттардың барлық деңгейлерін зерттеп, оларды құрастыру. Веб қосымшалардың өнімділігін арттыратын функцияларды құру және кодтың оқу барысын қадағалау.

Жұмыстың сипаттамасы. Жұмыс 2 бөліктен тұрады: