Идея на проекта:

### Проектът е на тема породи кучета от FCI(Federation Cynologique Internationale) и предоставя информация за различни породи кучета ,официално регистрирани към Международната Киноложка Асоциация(FCI).Самите XML файлове съдържат информация за няколко породи кучета,с които работя в проекта.Приложението от своя страна предоставя възможност за зареждане на набор от екземпляри(XML файлове) в база от данни разработена посредством Microsoft SQL Server 2008.Друга функционалност на приложението е ръчно попълване на данни за порода, след което се прави валидация на данните и при успешна такава се записва в създадената БД и едновременно с това се генерира XML файл съдържащ новопопълнената информация.

Съдържание на папката с файловете

* **ASP\_DTD.dtd** – Това е DTD файлът,който описва структурата на XML файловете(екземплярите)
* **ASP\_N\_xml\_file.xml** – N-тият XML екземпляр съдържащ данни.
* **K\_well\_formed\_test.xml** – К-тият грешно форматиран (сгрешен) xml файл.

**N € [first, second, third, fourth, fifth, sixth, seventh, eighth, ninth, tenth, eleventh, twelfth, thirteenth, fourteenth, fifteenth, sixteenth, seventeenth, eighteenth, nineteenth, twentieth ]**

**К € [first, second, third, fourth, fifth]**

Oписание на всички елементи и атрибути от DTD

<?xml version="1.0" encoding="UTF-8"?>  
<!ELEMENT fci ( dog\_breeds, sections, groups ) >  
 <!ELEMENT dog\_breeds (dog\_breed+)>  
 <!ELEMENT dog\_breed (name,country,year\_of\_establishment,head,teeth,ears,eyes,tail,colors,fur,image,size,weight)>  
 <!ATTLIST dog\_breed standart\_number ID #REQUIRED>  
 <!ATTLIST dog\_breed group IDREFS #REQUIRED>  
 <!ATTLIST dog\_breed section IDREFS #REQUIRED>  
 <!ELEMENT name (#PCDATA)>  
 <!ELEMENT country (country\_name,country\_continent,country\_government\_type)>  
 <!ATTLIST country capital CDATA #REQUIRED>  
 <!ATTLIST country official\_language CDATA #IMPLIED>  
 <!ATTLIST country time\_zone CDATA #IMPLIED>  
 <!ATTLIST country currency CDATA #IMPLIED>  
 <!ATTLIST country country\_code CDATA #IMPLIED>  
 <!ELEMENT country\_name (#PCDATA)>  
 <!ELEMENT country\_continent (#PCDATA)>  
 <!ELEMENT country\_government\_type (#PCDATA)>   
 <!ELEMENT year\_of\_establishment (#PCDATA)>  
 <!ELEMENT head (#PCDATA)>  
 <!ELEMENT teeth (#PCDATA)>  
 <!ELEMENT ears (#PCDATA)>  
 <!ELEMENT eyes (#PCDATA)>  
 <!ELEMENT tail (#PCDATA)>  
 <!ELEMENT colors (primary\_color,secondary\_color,prefered\_color)>  
 <!ELEMENT primary\_color (#PCDATA)>  
 <!ELEMENT secondary\_color (#PCDATA)>  
 <!ELEMENT prefered\_color (#PCDATA)>  
 <!ELEMENT fur (#PCDATA)>  
 <!ELEMENT image (#PCDATA)>  
 <!ELEMENT size (males\_size,females\_size)>  
 <!ELEMENT males\_size (#PCDATA)>  
 <!ELEMENT females\_size (#PCDATA)>  
 <!ELEMENT weight (males\_weight,females\_weight)>  
 <!ELEMENT males\_weight (#PCDATA)>  
 <!ELEMENT females\_weight (#PCDATA)>   
 <!ELEMENT sections (section+)>  
 <!ELEMENT section (#PCDATA)>  
 <!ATTLIST section section\_id ID #REQUIRED>  
 <!ELEMENT groups (group+)>  
 <!ELEMENT group (#PCDATA)>  
 <!ATTLIST group group\_code ID #REQUIRED >

## Дървовидно представяне на DTD

* fci
  + dog\_breeds
    - dog\_breed(standart\_number,group,section)
      * name
      * country(capital, official\_language, time\_zone, currency, country\_code)
        + country\_name
        + country\_continent
        + country\_government\_type
      * year\_of\_establishment
      * head
      * teeth
      * ears
      * eyes
      * tail
      * colors
        + primary\_color
        + secondary\_color
        + preferred\_color
      * fur
      * image
      * size
        + males\_size
        + females\_size
      * weight
        + males\_weigh
        + females\_weight
  + sections
    - section(section\_id)
  + groups
    - group(group\_code)

Ролята на кореновият елемент тук играе елементът : ***fci***  . Елементите ,които се съдържат в него са :   
***dog\_breeds –*** Съвкупност от породи кучета (***dog\_breed+***)

***Sections –*** Всяка порода кучета,официално регистрирана към **FCI** спада към определена секция на дадена група.Елементът ‘sections’ представя именно това-Информация за секцията ,към която приспада породата

***Groups-***Всяка порода кучета приспада към определена група.Групите са общо 10 наброй и сподпомагат структурирането и диференцирането на различните породи кучета в зависимост от техните поведенчески и екстериорни характеристики.Елементът ‘groups’ представя именно това.

***dog\_breeds*(Породи кучета)-**Състои се от поне един ***dog\_breed(*Опр. порода куче*).***

**dog\_breed(standart\_number,group,section)-**Съдържа следните елементи:  
 -**name-***Наименованието на породата*

-**country-***Държава основателка*

*-****country\_name*** *– Име на държавата основателка*

*-****country\_continent*** *– Континен, върху който попада държвата*

*-****country\_government\_type –****Тип на държавното управление*

**-year\_of\_establishment –** *Година на признаване на стандарта за породата*

**-head-***Информация за формата на главата на кучето*

**-teeth-***Информация за формата и структурата на зъбите на кучето*

**-ears-***Информация за формата на ушите на кучето*

**-eyes-***Информация за очите на кучето*

**-tail***-Информация за формата на опашката на кучето*

**-colors(primary\_color,secondary\_color,preferred\_color)-***Предоставя информация за окраската на кучето като ‚* **primary\_color,secondary\_color,preferred\_color** *‘ представят съответно първа възможна окраска,втора разновидност на окраската и предпочитана окраска*

**-fur-***Информация за козината на кучето*

**-image-***Изображение показващо примерен представител на породата*

**-size(males\_size,females\_size) –** *Този елемент представя информация за височината на кучето измерена при холката като ’* ***males\_size,females\_size****’ дават информация за височината съответно при мъжките и при женските екземпляри*

**-weight(males\_weigh,females\_weight)-***Това е елемент даващ информация за теглото на кучето и поделементите ‚* **males\_weigh,females\_weight‘** *дават информация съответно за теглото при мъжките и при женските представители на породата*

***!Всички елементи споменати по-горе са от тип #PCDATA***

**Атрибутите на dog\_breed : standart\_number,group,section ни дават информация за номера на стандарта на породата,групата ,към която спада тя и съответно секцията от съответната група (всяка група съдържа няколко секции).**

**standart\_number** - ID #REQUIRED типът на този атрибут е ID и е задължителен

**group -** IDREFS #REQUIRED този атрибут също е задължителен и е от тип IDREFS

**section** - IDREFS #REQUIRED този атрибут също е задължителен и е от тип IDREFS

**-sections –** Всеки елемент може да съдържа един или повече елемента от тип ***section***

като той сам по себе си има атрибут ***section\_id*** ID #REQUIRED

**-groups -** Всеки елемент може да съдържа един или повече елемента от тип ***group*** като той сам по себе си има атрибут **group\_code** ID #REQUIRED

**Атрибути на country : capital, official\_language, time\_zone, currency, country\_code Дават ни допълнителна информация за държавата основателка на породата като същевременно я идентифицират по уникален начин(Полезно за БД).**

capital - CDATA #REQUIRED - Типът на атрибута е CDATA и е задължителен.Дава информация за столицата на държавата.

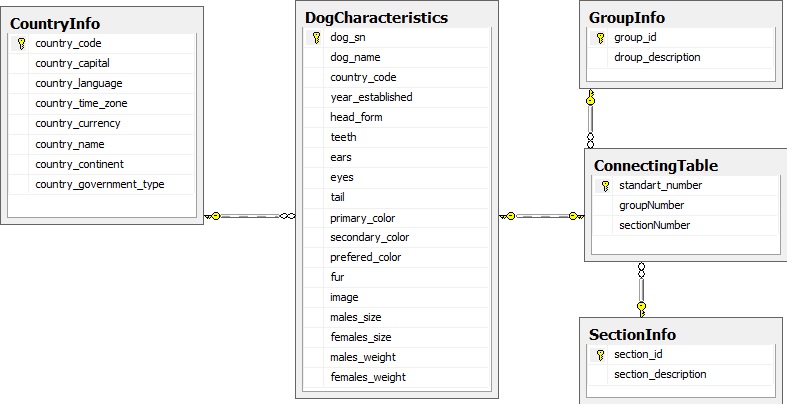
official\_language - CDATA #IMPLIED – Тип CDATA като този атрибут е от тип IMPLIED(подразбиращ се).Носи информация за официалният език.

time\_zone - CDATA #IMPLIED – Дава ни информация за часовата зона и типът му е същият като на по-горният атрибут.

currency - CDATA #IMPLIED – Дава информация за валутата в държавата основателка

country\_code - CDATA #IMPLIED – Обикновено това е 3-буквена абревиатура на името на държавата, записано на латиница.

Описание на всички таблици и колони от БД



**DogCharacteristics –** Тази таблица съдържа пълната информация (описание) за дадена порода кучета.Всички размери, цветове, тегла, окраски и т.н. се представят в тази таблица.Нейните колони са както следва:  
 **dog\_sn** – Номер на стандарта на породата

**dog\_name** – Име на породата

**country\_code** – Код на държавата основателка

**year\_established** – Година на основаване

**head\_form** – Форма на главата

**teeth** – Зъбна формула и челюсти

**ears** – Форма на ушите

**eyes** – Форма на очите

**tail** – Форма на опашката

**primary\_color** – Първа разцветка

**secondary\_color** – Втора разцветка

**preferred\_color** – Предпочитана разцветка

**fur** – Тип на козината

**image** – Изображение показващо примерен представител на породата

**males\_size** – Размери при мъжките екземпляри

**females\_size** – Размери при женските екземпляри

**males\_weight** – Тегло при мъжките

**females\_weight** – Тегло при женските

**CountryInfo** - Тази таблица съдържа изчерпателна информация свързана с държавата основателка на конкретната порода кучета.

**country\_code** – Код на държавата основателка

**country\_capital** – Столица на държавата основателка

**country\_language** – Официален език

**country\_time\_zone** – Часова зона, в която попада държавата основателка(За големите държави се взима конкретен град)

**country\_currency** – Парична единица

**country\_name** – Същинското име на държавата основателка

**country\_continent** – Континент върху, който попада държавата.

**country\_government\_type** – Тип на държавното управление

**ConnectingTable** - Тази таблица играе ролята на свързваща.Чрез нея се избягват излишествата, които биха могли да се породят я в таблиците свързани с описанието на секцията и групата, към която приспада породата.

**standart\_number** – Номер на стандарта

**groupNumber** – Номер на групата

**sectionNumber** – Номер на секцията в дадена група, към която приспада породата

**GroupInfo** - Тази таблица съдържа изчерпателна информация свързана с държавата групата, към която приспада конкретната порода кучета.

**group\_id** – Номер на група

**group\_description** – Описание на групата

**SectionInfo** - Тази таблица съдържа изчерпателна информация свързана със секцията, към която приспада конкретната порода кучета.

section\_id – Номер на секцията

section\_description – Описание на секцията

Информация за това, кой елемент(атрибут) от DTD в коя таблица и съответно коя колона се прехвърля след зареждането на файловете в Базата данни.

**Attr : standart\_number** - >В таблиците DogCharacteristics и ConnectingTable съответно в колоните : dog\_sn , standart\_number

**Attr : group** ->В таблиците ConnectingTable И GroupInfo съответно в колоните: groupNumber, group\_id

**Attr : section** ->В таблиците ConnectingTable И SectionInfo съответно в колоните: sectionNumber, section\_id

**Elem : name** ->В таблицата DogCharacteristics , в колоната dog\_name

**Attr : country\_capital** ->В таблицата CountryInfo, в колоната country\_capital

**Attr : official\_language** ->В таблицата CountryInfo, в колоната country\_language

**Attr : time\_zone** ->В таблицата CountryInfo, в колоната country\_time\_zone

**Attr : currency** -> В таблицата CountryInfo, в колоната country\_currency

**Attr : country\_code** -> В таблицата CountryInfo, в колоната country\_code

**Elem : country\_name** -> В таблицата CountryInfo, в колоната country\_name

**Elem : country\_continent** -> В таблицата CountryInfo, в колоната country\_continent

**Elem:country\_government\_type** -> В таблицата CountryInfo, в колоната country\_government\_type

**Elem : year\_of\_establishment** ->В таблицата DogCharacteristics, в колоната year\_established

**Elem : head** -> В таблицата DogCharacteristics, в колоната head\_form

**Elem : teeth** -> В таблицата DogCharacteristics, в колоната teeth

**Elem : ears** -> В таблицата DogCharacteristics, в колоната ears

**Elem : eyes** -> В таблицата DogCharacteristics, в колоната eyes

**Elem : tail** -> В таблицата DogCharacteristics, в колоната tail

**Elem : primary\_color** -> В таблицата DogCharacteristics, в колоната primary\_color

**Elem : secondary\_color** -> В таблицата DogCharacteristics, в колоната secondary\_color

**Elem : prefered\_color** -> В таблицата DogCharacteristics, в колоната primary\_color

**Elem : fur** -> В таблицата DogCharacteristics, в колоната fur

**Elem : image**  -> В таблицата DogCharacteristics, в колоната image

**Elem : males\_size** -> В таблицата DogCharacteristics, в колоната males\_size

**Elem : females\_size** -> В таблицата DogCharacteristics, в колоната females\_size

**Elem : males\_weight** -> В таблицата DogCharacteristics, в колоната males\_weight

**Elem : females\_weight** -> В таблицата DogCharacteristics, в колоната females\_weight

**Elem : section(от sections)** ->В таблицата SectionInfo, в колоната section\_description

**Elem : group(от groups)** ->В таблицата GroupInfo, в колоната group\_description

Описание на функцията, чрез която съдържанието на 20-те XML файла се прехвърля в базата данни.

***Това е функцията, чрез която, след натискането на бутона “Въвеждане на XML документите в БД” всички XML файлове се валидират и съответно вкарват/не се вкарват в Базата данни. За всеки един ред има обяснение какво прави и каква му е целта:***

***Двата основни параметъра, с които работи функцията идват от класа и те са :***

readonly string[] \_xmlContainer = Directory.GetFiles(@"D:\\Кико\\FMI STUFF\\ASP\_Project\\XML\_Files","\*.xml"); -> Мястото, където локално се съхраняват 20-те XML файла.

**errorsLog\_TextBox** – Това е всъщност прозореца, на който се изобразява цялата информация свързана : с валидация, дублиране и съответно успешно/неуспешно въвеждане в БД.

***Функцията:***

protected void submit\_Button\_Click(object sender, EventArgs e)

{

//Тук изчистваме текстовото поле, в което се показва резултатът от въвеждането на данните в БД

submissionResult\_TextBox.Text = string.Empty;

for (var k = 0; k < \_xmlContainer.Length; k++)

{

if (RecordsCheckAndValidation.ValidationResult(\_xmlContainer[k], errorsLog\_TextBox))

{

var existsOrNotInDb = "Да";

//Създаване на нов XML document

var reader = new XmlDocument();

reader.Load(\_xmlContainer[k]);

//С този ред запазваме в променливата elementList всички поделементи на dog\_breed като група

XmlNodeList elementsList = reader.GetElementsByTagName("dog\_breed");

//Тук в countryList запазваме списък от групи с (country\_name,country\_continent,country\_government\_type..)

XmlNodeList countryList = reader.GetElementsByTagName("country");

//В тази променлива запазваме групите свързани с (primary\_color,secodnary\_color,prefered\_color)

XmlNodeList colorsList = reader.GetElementsByTagName("colors");

//В този списък запазваме групите свързани със (males\_size,females\_size)

XmlNodeList sizesList = reader.GetElementsByTagName("size");

//В този списък запазваме групите свързани със (males\_weight,females\_weight)

XmlNodeList weightsList = reader.GetElementsByTagName("weight");

//В този списък запазваме стойностите, които стоят в елемента <groups>

XmlNodeList seperateGroupsList = reader.GetElementsByTagName("groups");

//В този списък запазваме стойностите, които стоят в елемента <sections>

XmlNodeList seperateSectionList = reader.GetElementsByTagName("sections");

for (int i = 0; i < elementsList.Count; i++)

{

//Тук е мястото, където ще създадем новият обект от тип dogRecord

//Който ще попълваме в следващите стъпки от тази итерация.

//След като го пъплним на всяка една от стъпките, след това би трябвало да ни е по-лесно да

//работм с него при добавянето в БД

dogRecord tmpDog = new dogRecord();

foreach (XmlElement element in elementsList[i].ChildNodes)

{

if (element.HasAttributes)

{

//На този ред прочитаме атрибутите на dog\_breed(standart\_number,group,section)

foreach (XmlAttribute atr in element.ParentNode.Attributes)

{

//richTextOutput\_TextBox.Text += atr.Name + "::" + atr.InnerText + "\n";

if (atr.Name == "standart\_number") { tmpDog.standart\_number=atr.InnerText; }

if (atr.Name == "group") { tmpDog.group = atr.InnerText; }

if (atr.Name == "section") { tmpDog.section = atr.InnerText; }

//На този ред започва сравненеито между лемемнтите в <groups> и текущата group за да можем да

//вземем описанието на групата, в която попада породата куче

for (int j = 0; j < seperateGroupsList.Count; j++)

{

foreach (XmlElement groupElement in seperateGroupsList[j].ChildNodes)

{

//richTextOutput\_TextBox.Text += groupElement.InnerText;

if (groupElement.HasAttribute("group\_code") && atr.InnerText == groupElement.GetAttribute("group\_code"))

{

//richTextOutput\_TextBox.Text += "Group Description"+"::"+groupElement.InnerText + "\n";

tmpDog.group\_description = groupElement.InnerText;

}

}

}

//На този ред започва сравненеито между елементите в <sections> и текущата section за да можем да

//вземем описанието на секцията, в която попада породата куче

for (int j = 0; j < seperateSectionList.Count; j++)

{

foreach (XmlElement sectionElement in seperateSectionList[j].ChildNodes)

{

if (sectionElement.HasAttribute("section\_id") && atr.InnerText == sectionElement.GetAttribute("section\_id"))

{

tmpDog.section\_description = sectionElement.InnerText;

}

}

}

}

//Тук прочитаме атрибутите на country(country\_code,capital,official\_language,time\_zone,currency)

foreach (XmlAttribute el in element.Attributes)

{

if (el.Name == "country\_code") { tmpDog.country\_code = el.InnerText; }

if (el.Name == "capital") { tmpDog.capital=el.InnerText; }

if (el.Name == "official\_language") { tmpDog.official\_language = el.InnerText; }

if (el.Name == "time\_zone") { tmpDog.time\_zone = el.InnerText; }

if (el.Name == "currency") { tmpDog.currency = el.InnerText; }

}

}

if (element.Name != "country" && element.Name != "colors" && element.Name != "size" && element.Name != "weight")

{

if (element.Name == "name"){tmpDog.name = element.InnerText;}

if (element.Name == "year\_of\_establishment")

{ tmpDog.year\_establishment= Convert.ToInt32(element.InnerText); }

if (element.Name == "head") { tmpDog.head = element.InnerText; }

if (element.Name == "teeth") { tmpDog.teeth = element.InnerText; }

if (element.Name == "ears") { tmpDog.ears = element.InnerText; }

if (element.Name == "eyes") { tmpDog.eyes = element.InnerText; }

if (element.Name == "tail") { tmpDog.tail = element.InnerText; }

if (element.Name == "fur") { tmpDog.fur = element.InnerText; }

if (element.Name == "image") { tmpDog.image = element.InnerText; }

}

if (element.Name == "country")

{

foreach (XmlElement cElement in countryList[i].ChildNodes)

{

if (cElement.Name == "country\_name") {tmpDog.country\_name=cElement.InnerText; }

if (cElement.Name == "country\_continent") { tmpDog.country\_continent=cElement.InnerText; }

if (cElement.Name == "country\_government\_type") { tmpDog.country\_government\_type=cElement.InnerText; }

}

}

if (element.Name == "colors")

{

foreach (XmlElement colElement in colorsList[i].ChildNodes)

{

if (colElement.Name == "primary\_color") { tmpDog.primary\_color=colElement.InnerText; }

if (colElement.Name == "secondary\_color") { tmpDog.secondary\_color = colElement.InnerText; }

if (colElement.Name == "prefered\_color") { tmpDog.prefered\_color = colElement.InnerText; }

}

}

if (element.Name == "size")

{

foreach (XmlElement sizeElement in sizesList[i].ChildNodes)

{

if (sizeElement.Name == "males\_size") { tmpDog.males\_size=sizeElement.InnerText; }

if (sizeElement.Name == "females\_size") { tmpDog.females\_size = sizeElement.InnerText; }

}

}

if (element.Name == "weight")

{

foreach (XmlElement weightElement in weightsList[i].ChildNodes)

{

if (weightElement.Name == "males\_weight") { tmpDog.males\_weight = weightElement.InnerText; }

if (weightElement.Name == "females\_weight") { tmpDog.females\_weight = weightElement.InnerText; }

}

}

}

//Преди да вмъкнем записа проверяваме дали вече има такъв в БД и съответно извеждаме съобщение дали го има

//Или го няма в БД

if (RecordsCheckAndValidation.IsThereSuchAStandartNumber(tmpDog.standart\_number))

{

existsOrNotInDb = "Вече съществува такъв запис в базата данни";

}

//Тук е мястото, на което след като вече имаме dogRecord попълнен го подаваме на функция

// insertAdogRecordIntoTheDataBase(dogRecord object) и тя вече преценява, какво да прави с него, тоест

//Дали е валиден за въвеждане.Ако - да го въвежда, ако- не, не го въвежда и преминаваме нататък

InsertDogBreedRecord.InsertAdogRecordIntoTheDataBase(tmpDog,errorsLog\_TextBox);

}

submissionResult\_TextBox.Text += "(" + k + ")";

submissionResult\_TextBox.Text +=("Име на файл:"+Path.GetFileName(\_xmlContainer[k]) + "\n \n Валиден:ДА\n Добавен към базата данни:"+existsOrNotInDb+"\n");

}

else

{

submissionResult\_TextBox.Text += "(" + k + ")";

submissionResult\_TextBox.Text +="Име на файл:"+ Path.GetFileName(\_xmlContainer[k]) + "\n \n Валиден:НЕ \n Добавен в базата данни:НЕ \n";

}

submissionResult\_TextBox.Text += "\n ================================================= \n";

}

}

***Това е функцията, чрез която, след натискането на бутона “Попълване на XML документ” , съдържанието на ръчно въведен XML документ се валидира за БД и съответно се добавя/не се добавя в нея.След това се генерира XML документ отговарящ на току-що попълненият документ.***

protected void submitXmlButton\_Click ( object sender, EventArgs e )

{

//Тук изчистваме резултата преди всяко повторно натискане на бутона

//за да избегнем натупването на надписи

submitResult\_Lable.Text = string.Empty;

standratNumberStatus\_Label.Text = string.Empty;

groupStatus\_Label.Text = string.Empty;

sectionStatus\_Label.Text = string.Empty;

countryCodeStatus\_Label.Text = string.Empty;

yearEstablishedStatus\_Label.Text = string.Empty;

//Тук изивкваме помощната функция

SubmitXmlButtonHelper();

}

***След това се извиква функцията:***

private void SubmitXmlButtonHelper ( )

{

//Тук се прави проверката ако всичко е валидно в току-що запълнените полета – праща се в БД и се изкарва съобщение за успешно добавен запис.В противен случай се изкарват надписи до полетата, които не са се валидирали правилно или просто се изкарва съобщение, че вече съществува такъв запис в БД

if (RecordsCheckAndValidation.IsStandartNumberValid(standartNumber\_Input)

&& RecordsCheckAndValidation.IsGroupNumberValid(groupInput)

&&RecordsCheckAndValidation.IsSectionNumberValid(sectionInput,groupInput)

&&RecordsCheckAndValidation.IsCountryCodeNumberValid(countryCodeInput))

{

//ФУнкцията, която извършва добавянето

ButtonClickHandler.SubmitXmlButtonWithSuccessfulValidation(standratNumberStatus\_Label,groupStatus\_Label,sectionStatus\_Label,

countryCodeStatus\_Label,yearEstablishedStatus\_Label,submitResult\_Lable,

standartNumber\_Input,groupInput,sectionInput,groupDescrInput,sectionDescrInput,

breedNameInput,countryCodeInput,countryCapitalInput,officialLanguageInput,

timeZoneInput,currencyInput,countryNameInput,countryContinent\_DropDownListInput,

governmentTypeInput,yearEstablishedInput,headFormInput,teethInput,earsInput,

eyesInput,tailInput,primaryColorInput,secondaryColorInput,preferedColorInput,

furInput,imageInput,malesSizeInput,femalesSizeInput,malesWeightInput,

femalesWeightInput,submitResult\_Lable,errorsLog\_TextBox);

}

//В противен случай се извеждат надписи до текстовите полета, в които е имало проблем при валидацията

else

{

submitResult\_Lable.Text = "Получи се проблем при валидацията на въведените данни.";

submitResult\_Lable.Visible = true;

submitResult\_Lable.ForeColor = System.Drawing.Color.Red;

submitResult\_Lable.Dispose ( );

if (!RecordsCheckAndValidation.IsStandartNumberValid(standartNumber\_Input))

{

standratNumberStatus\_Label.Text = "Въведеният номер на стандарт е невалиден!";

standratNumberStatus\_Label.Visible = true;

standratNumberStatus\_Label.ForeColor = System.Drawing.Color.IndianRed;

standratNumberStatus\_Label.Dispose ( );

}

if (!RecordsCheckAndValidation.IsGroupNumberValid(groupInput))

{

groupStatus\_Label.Text = "Въведеният номер на група е невалиден!";

groupStatus\_Label.Visible = true;

groupStatus\_Label.ForeColor = System.Drawing.Color.IndianRed;

groupStatus\_Label.Dispose ( );

}

if (!RecordsCheckAndValidation.IsSectionNumberValid(sectionInput,groupInput))

{

sectionStatus\_Label.Text = "Въведеният номер на секция е невалиден!";

sectionStatus\_Label.Visible = true;

sectionStatus\_Label.ForeColor = System.Drawing.Color.IndianRed;

sectionStatus\_Label.Dispose();

}

if (!RecordsCheckAndValidation.IsCountryCodeNumberValid(countryCodeInput))

{

countryCodeStatus\_Label.Text = "Въведеният Код на държава е невалиден!";

countryCodeStatus\_Label.Visible = true;

countryCodeStatus\_Label.ForeColor = System.Drawing.Color.IndianRed;

countryCodeStatus\_Label.Dispose();

}

if (!RecordsCheckAndValidation.IsYearEstablishedValid(yearEstablishedInput))

{

yearEstablishedStatus\_Label.Text = "Въведената година е невалидна!";

yearEstablishedStatus\_Label.Visible = true;

yearEstablishedStatus\_Label.ForeColor = System.Drawing.Color.IndianRed;

yearEstablishedStatus\_Label.Dispose();

}

}

}

***Това е помощната функция за въвеждане на записа в БД:***

public static void SubmitXmlButtonWithSuccessfulValidation( Label snStatusLabel,Label groupStatusLabel,Label sectionStatusLabel,

Label countryCodeStatusLabel,Label yearEstablishedStatusLabel,Label submitResultLabel,

TextBox snInput,TextBox grInput,TextBox secInput,TextBox grDescrInput,

TextBox secDescrInput,TextBox brNameInput,TextBox cntryCodeInput,

TextBox cntrCapInput,TextBox offLangInput,DropDownList tZoneInput,

TextBox currInput,TextBox cntrNameInput,DropDownList cntryContinentInput,

TextBox govTypeInput,TextBox yearEstInput,TextBox headFrmInput,TextBox teethFormInput,

TextBox earsFormInput,TextBox eyesFormInput, TextBox tailFormInput,

TextBox primColInput,TextBox secColInput,TextBox prefColInput,TextBox furFormInput,

TextBox imgInput,TextBox mSizeInput,TextBox fSizeInput,

TextBox mWeghtInput,TextBox fWeghtInput,Label sbmtResLabel,TextBox errTextBox)

{

snStatusLabel.Text = string.Empty;

groupStatusLabel.Text = string.Empty;

sectionStatusLabel.Text = string.Empty;

countryCodeStatusLabel.Text = string.Empty;

yearEstablishedStatusLabel.Text = string.Empty;

submitResultLabel.Text = "Записът е валиден и ще бъде добавен в базата данни!";

submitResultLabel.Visible = true;

submitResultLabel.ForeColor = System.Drawing.Color.Azure;

submitResultLabel.Dispose();

if (!RecordsCheckAndValidation.IsThereSuchAStandartNumber(snInput.Text))

{

//Тук след като вече сме проверили,че ключовите полета са валидни

//създаваме запис от тип dogRecord и го въвеждаме в базата данни

var myDogRec = new dogRecord(snInput.Text, grInput.Text, secInput.Text,

grDescrInput.Text, secInput.Text, brNameInput.Text,

cntryCodeInput.Text, cntrCapInput.Text, offLangInput.Text,

tZoneInput.Text, currInput.Text, cntrNameInput.Text,

cntryContinentInput.Text, govTypeInput.Text,

yearEstInput.Text != string.Empty

? Convert.ToInt32(yearEstInput.Text)

: 0, headFrmInput.Text, teethFormInput.Text, earsFormInput.Text, eyesFormInput.Text,

tailFormInput.Text, primColInput.Text, secColInput.Text,

prefColInput.Text, furFormInput.Text, imgInput.Text, mSizeInput.Text,

fSizeInput.Text, mWeghtInput.Text, fWeghtInput.Text);

InsertDogBreedRecord.InsertAdogRecordIntoTheDataBase(myDogRec, errTextBox );

//След това създаваме XML документ, валиден за посоченото DTD и го попълваме с въведените данни

InsertDogBreedRecord.CreateValidXmlDocument(snInput,grInput,secInput,brNameInput,cntrCapInput,

offLangInput,tZoneInput,currInput,cntryCodeInput,

cntrNameInput,cntryContinentInput,govTypeInput,

yearEstInput,headFrmInput,teethFormInput,earsFormInput,

eyesFormInput,tailFormInput,primColInput,secColInput,

prefColInput,furFormInput,imgInput,mSizeInput,fSizeInput,

mWeghtInput,fWeghtInput,secDescrInput,grDescrInput);

}

else

{

sbmtResLabel.Text =

"Записът е валиден \n Но няма да бъде въведен в БД защото вече същестува такъв запис!";

sbmtResLabel.ForeColor = System.Drawing.Color.Lime;

sbmtResLabel.Dispose();

}

}



Софийски университет

Св.Климент Охридски

# ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

#### ДОКУМЕНТАЦИЯ

КЪМ КУРСА „ASP ПРОГРАМИРАНЕ“

КЪМ ПРОЕКТ „Породи кучета официално регистрирани към FCI Международна Киноложка Асоциация“

Изготвил:

**Кристиан Пеев Азманов , ФН:71323**

**Информационни системи ,3 курс,2-ра група**

**01.06.2013**