|  |  |
| --- | --- |
| SU_logo.png | **Софийски университет “Св. Климент Охридски”**  **Факултет по математика и информатика** |

**ДОКУМЕНТАЦИЯ**

към проект по „ASP програмиране“

Тема: Симулация на движението на роботи по карти.

|  |  |
| --- | --- |
| **Изготвил:**  ***Антон Дудов, ФН: 71488, ИС курс 3*** | **Под ръководството на:**  ***доц. д-р Павел Павлов*** |

София 2016

**Съдържание:**

1. **Описание на проекта.**
2. **Файлове в проекта и описание за тях.**
3. **DTD (Robosim.dtd).**
4. **Описание на елементите на DTD (Robosim.dtd).**
5. **Описание на атрибутите на DTD (Robosim.dtd).**
6. **Описание на базата от данни (БД).**
7. **Схема на взаимовръзката между таблиците на БД.**
8. **Прехвърляне на информацията от XML към БД.**
9. **Приложение, работещо с XML файловете и БД.**
10. **Описание на проекта.**

Проектът представя структурата на роботи, карти и алгоритми за движението на роботите по съответните карти. Описано е:

* Симулация
* Роботи
* Карти
* Околни среди
* Алгоритми

В „Симулация“ има информация за симулацията, името, собственика, емайл на собственика, рейтинг, описание/белешки.

В „Роботи“ има информация за всеки робот. Геометрията на робота, какви сензори има, разположение на колелата, перки, информация за други подвижни части, максимални скорости.

В „Карти“ има информация за всяка карта. Пътят до двоичния файл, в който са данните за картата, информация за денивелацията и името на картата.

В „Околни среди“ има информация за средите(околната среда). Всяка околна среда има име и стойност, която показва с какво „усилие“ робота преминава през нея за единица време. (Например такива среди биха били: суша, въздух(ако трябва да бъде „прескочена“ някоя пропаст или стена) и вода.

В „Алгоритми“ има информация за всеки алгоритъм. Данните за алгоритъма са името, сложността му, дали поддържа множество от целеви точки и дали може да прави разлика между различни околни среди(например- през вода да преминава по-бавнo, отколкото по суша). Използва се за определяне на кой алгоритъм за търсене да бъде пуснат.

1. **Файлове в проекта и описание за тях.**

* RoboSim.dtd – Файл, съдържащ описанието на елементите и атрибутите, които се съдържат в XML документите. Съдържанието му е приложено по-нататък в документацията.
* ValidXMLDocument-X.xml – XML документите, които са валидирани спрямо RoboSim.dtd. В името на файла „Х“ представлява номера на документа и приема стойности от 1 до 20.
* AddXMLFiles.aspx – Уеб страница базирана на ASP.NET която проверява валидността на XML документите и записва в базата данни информацията съдържаща се във валидните от тях. Страницата визуализира резултатите от валидацията и запазването на информацията.
* Utility.cs – файл написн със средствата на езика С#, който извършва същинската валидация и запис в базата от данни.
* EnterData.aspx – Уеб страница базирана на ASP.NET, която съдържа в себе си форма за въвеждане на информация от потребителя. Освен това тя записва въведената информация в нов XML файл и в базата от данни.

1. **DTD (RoboSim.dtd).**

<?xml version="1.0" encoding="UTF-8"?>  
<!ELEMENT RoboSimulation (SimulationOwnerEmail, SimulationDescription, SimulationRating, Environments, Robots, Maps, Algorithms)>  
 <!ATTLIST RoboSimulation id ID #REQUIRED   
 name CDATA #REQUIRED  
 owner CDATA #REQUIRED>  
 <!ELEMENT SimulationOwnerEmail (#PCDATA)>  
 <!ELEMENT SimulationDescription (#PCDATA)>  
 <!ELEMENT SimulationRating (#PCDATA)>   
 <!ELEMENT Environments (Environment\*)>  
 <!ELEMENT Environment (TravelCostEnter, TravelCostIn, TravelCostExit, Damage)>  
 <!ATTLIST Environment id ID #REQUIRED  
 name CDATA #REQUIRED>  
 <!ELEMENT TravelCostEnter (#PCDATA)>  
 <!ELEMENT TravelCostIn (#PCDATA)>  
 <!ELEMENT TravelCostExit (#PCDATA)>  
 <!ELEMENT Damage (#PCDATA)>  
 <!ELEMENT Robots (Robot\*)>  
 <!ELEMENT Robot (RobotMeshGrid, Speed, SpeedBack, TurningSpeed, TurningSpeedBack, Wheels?, Sensors?, Rotors?)>  
 <!ATTLIST Robot id ID #REQUIRED  
 name CDATA #REQUIRED  
 owner CDATA #REQUIRED>  
 <!ELEMENT RobotMeshGrid (#PCDATA)>  
 <!ELEMENT Speed (#PCDATA)>  
 <!ELEMENT SpeedBack (#PCDATA)>  
 <!ELEMENT TurningSpeed (#PCDATA)>  
 <!ELEMENT TurningSpeedBack (#PCDATA)>   
 <!ELEMENT Wheels (Wheel\*)>   
 <!ELEMENT Wheel (WheelMeshGrid, WheelDiameter, WheelWidth)>  
 <!ATTLIST Wheel driving (ДА|НЕ) #REQUIRED>  
 <!ELEMENT WheelMeshGrid (#PCDATA)>  
 <!ELEMENT WheelDiameter (#PCDATA)>  
 <!ELEMENT WheelWidth (#PCDATA)>  
 <!ELEMENT Sensors (Sensor\*)>  
 <!ELEMENT Sensor (SensorMeshGrid, NumberOfValusPerSecond)>  
 <!ATTLIST Sensor name CDATA #REQUIRED  
 valueType CDATA #REQUIRED>  
 <!ELEMENT SensorMeshGrid (#PCDATA)>  
 <!ELEMENT NumberOfValusPerSecond (#PCDATA)>  
 <!ELEMENT Rotors (Rotor\*)>  
 <!ELEMENT Rotor (RotorMeshGrid, RotorLiftingPower)>  
 <!ELEMENT RotorMeshGrid (#PCDATA)>  
 <!ELEMENT RotorLiftingPower (#PCDATA)>  
 <!ELEMENT Maps (Map\*)>  
 <!ELEMENT Map (MapData, Denivelation)>  
 <!ATTLIST Map id ID #REQUIRED  
 name CDATA #REQUIRED>  
 <!ELEMENT MapData (#PCDATA)>  
 <!ELEMENT Denivelation (#PCDATA)>  
 <!ELEMENT Algorithms (Algorithm\*)>  
 <!ELEMENT Algorithm (Complexity, Depth)>  
 <!ATTLIST Algorithm id ID #REQUIRED  
 name CDATA #REQUIRED   
 diffEnvironments (ДА|НЕ) #REQUIRED  
 multipleDestPoints (ДА|НЕ) #REQUIRED>  
 <!ELEMENT Complexity (#PCDATA)>  
 <!ELEMENT Depth (#PCDATA)>

1. **Описание на елементите на DTD (RoboSim.dtd).**
2. RoboSimulation – основен елемент на всеки XML документ(кореновият елемент).
3. SimulationOwnerEmail – съдържа информация за пощата на собственика на симулацията.
4. SimulationDescription – съдържа информация за описанието на симулацията.
5. SimulationRating – съдържа информация за рейтинга на симулацията.
6. Environments – обединява всички околни среди.
7. Environment – съдържа информация за конкретна околна среда.
8. TravelCostEnter – съдържа информация за „цената“ за влизане в конкретната околна среда.
9. TravelCostIn – съдържа информация за „цената“ за преминаване на единица разстояние в конкретната околна среда.
10. TravelCostOut – съдържа информация за „цената“ за излизане от конкретната околната среда.
11. Damage – съдържа информация за това какво количество щети ще поеме робота, ако прекара единица време в тази конкретна околна среда.
12. Robots – обединява всички роботи.
13. Robot – съдържа информация за конкретен робот.
14. RobotMeshGrid – съдържа адреса на файла, в който е информацията за геометрията на робота.
15. Speed – съдържа информация за скоростта на робота при движение в права линия.
16. SpeedBack – съдържа информация за скоростта на робота при движение в права линия назад.
17. TurningSpeed – съдържа информация за скоростта на робота при завиване.
18. TurningSpeedBack – съдържа информация за скоростта на робота при завиване назад.
19. Wheels – обединява всички колела на робота.
20. Wheel – съдържа информация за конкретно колело.
21. WheelMeshGrid – съдържа адреса на файла, в който е информацията за геометрията на колелото.
22. WheelDiameter – съдържа информация за диаметъра на колелото. С нея ще се пресметне какво разстояние ще премине робота за едно завъртане на колелото.
23. WheelWidth – съдържа информация за ширината на колелото.
24. Sensors – обединява всички сензори на робота.
25. Sensor – съдържа информация за конкретен сензор.
26. SensorMeshGrid – съдържа адреса на файла, в който е информацията за геометрията на сензора.
27. NumberOfValusPerSecond– съдържа информация за това колко измервания и съответно върнати стойности може да отчете сензора за една секунда.
28. Rotors – обединява всички перки на робота.
29. Rotor– съдържа информация за конкретна перка.
30. RotorMeshGrid - съдържа адреса на файла, в който е информацията за геометрията на перката.
31. RotorLiftingPower – съдържа информация за подемната сила на перката. С нея ще се пресметне какво разстояние ще измине за единица оборот на перката.
32. Maps – обединява всички карти.
33. Map – съдържа информация за конкретна карта.
34. MapData – съдържа адреса на файла, в който е информацията за самата карта.
35. Denivelation – съдържа инфрмация за максималната денивелация по картата в градуси.
36. Algorithms – обединява всички алгоритми за търсене по картите.
37. Algorithm – съдържа информация за конкретен алгоритъм.
38. Complexity – съдържа информация за времевата сложност, за която може да бъде намерен път между робота и целта му.
39. Depth – съдържа информация за това на каква дълбочина максимално да се пуска алгоритъма. Т.е. ако е много голяма картата може на етапи да се търси пътя от робота до целта му.
40. **Описание на атрибутите на DTD (RoboSim.dtd).**
41. id – атрибут на елемента RoboSimulation. Показва идентификационният номер на симулацията.
42. name – атрибут на елемента RoboSimulation. Показва името на симулацията.
43. owner – атрибут на елемента RoboSimulation. Показва собственика на симулацията.
44. name – атрибут на елемента Environment. Показва името на околната среда.
45. id – атрибут на елемента Environment. Показва идентификационният номер на всяка от околните среди.
46. id – атрибут на елемента Robot. Показ ва идентификационният номер на всеки от роботите.
47. name – атрибут на елемента Robot. Показва името на робота.
48. Owner – атрибут на елемента Robot. Показва собственика на робота.
49. driving – атрибут на елемента Wheel. Показва дали конкретното колело на робота е задвижващо или не. (ДА|НЕ) стойности.
50. name – атрибут на елемента Sensor. Показва името на робота.
51. valueType – атрибут на елемента Sensor. Показва каква величина измерва сензора(например температура, разстояние до обект и т.н.).
52. id – атрибут на елемента Map. Показва идентификационният номер на всяка от картите.
53. name – атрибут на елемента Map. Показва името на картата.
54. id – атрибут на елемента Algorithm. Показва идентификационният номер на всеки от алгоритмите.
55. name – атрибут на елемента Algorithm. Показва името на алгоритъма.
56. diffEnvironments – атрибут на елемента Algorithm. Показва дали алгоритъма поддържа различаване на околните среди. (ДА|НЕ) стойности.
57. multipleDestPoints – атрибут на елемента Algorithm. Показва дали алгоритъма поддържа откриване на максимален път до множество целеви точки. (ДА|НЕ) стойности.

Всички атрибути са задължителни.

1. **Описание на базата от данни (БД).**

Базата от данни съдържа 15 таблици

* **RoboSimulations** – съдържа основната информация за симулациите – име на симулацията, име на собственик, емайл на собственика, описание и оценка(от 0.00 до 6.00).
* **Environments** – съдържа информация за околните среди – име, цени за преминаване и поети щети.
* **Robots** – съдържа основната информация за роботите – име, собственик, геометрия и скорости.
* **Wheels** – съдържа информация за колелата – дали е задвижващо, геометрия, диаметър и широчина.
* **Sensors –** съдържа информация за сензорите – име, тип на отчетената стойност, геометрия и брой измервания за секунда.
* **Rotors –** съдържа информация за витлата – геометрия и повдигаща мощ.
* **Maps –** съдържа информация за картите – име, файл и максималната денивелация.
* **Algorithms –** съдържа информация за алгоритмите, с които се търси оптимален път между начална и крайна(крайни) точки по картата – име, дали поддържа различни околни среди и крайни точки на търсене, сложност и максимална дълбочина.

RoboSimulationsToEnvironments, RoboSimulationsToRobots, RoboSimulationsToMaps, RoboSimulationsToAlgorithms, RobotsToWheels, RobotsToSensors и RobotsToRotors са свързващи таблици.

Таблицата **RoboSimulations** съдържа следните колони:

* id – уникален идентификатор на симулацията. От тип int. Първичен ключ на таблицата. Генерира се автоматично от базата.
* simulation\_name – име на симулацията. От тип nvarchar(64).
* simulation\_owner – име на собственика. От тип nvarchar(64).
* simulation\_owner\_email – емайл на собственика. От тип nvarchar(32).
* simulation\_description – описание на симулацията. От тип nvarchar(1024).
* simulation\_rating – оценка на симулацията. От тип decimal(16,2). Има ограничение да е между 0.00 и 6.00 включително.

Таблицата **Environments** съдържа следните колони:

* id – уникален идентификатор на околната среда. От тип int. Първичен ключ на таблицата. Генерира се автоматично от базата.
* travel\_cost\_enter – цена за влизане в поле на тази околна среда. От тип decimal(16,2). Има ограничение да е положително или нула.
* travel\_cost\_in – цена за преминаване от едно в друго поле на тази околна среда. От тип decimal(16,2). Има ограничение да е положително или нула.
* travel\_cost\_ exit – цена излизане от поле на тази околна среда. От тип decimal(16,2). Има ограничение да е положително или нула.
* damage – поети щети за единица време престой тази околна среда. От тип decimal(16,2). Има ограничение да е положително или нула.

Таблицата **Robots** съдържа следните колони:

* id – уникален идентификатор на робота. От тип int. Първичен ключ на таблицата. Генерира се автоматично от базата.
* name – име на робота. От тип nvarchar(64).
* owner – име на собственика на робота. От тип nvarchar(64).
* robot\_mesh\_grid – геометрия на робота(файлово име). От тип nvarchar(32).
* speed – максимална скорост при движение напред. От тип decimal(16,2). Има ограничение да е положително или нула.
* speed\_back– максимална скорост при движение назад. От тип decimal(16,2). Има ограничение да е положително или нула.
* turning\_speed – максимална скорост на завиване при движение напред. От тип decimal(16,2). Има ограничение да е положително или нула.
* turning\_speed\_back– максимална скорост на завиване при движвение назад. От тип decimal(16,2). Има ограничение да е положително или нула.

Таблицата **Wheels** съдържа следните колони:

* id – уникален идентификатор на колелото. От тип int. Първичен ключ на таблицата. Генерира се автоматично от базата.
* driving – дали колелото е задвижващо или не. От тип nvarchar(2). Има ограничение да е една от следните две думи- “ДА” или “НЕ”.
* wheel\_mesh\_grid – геометрия на колелото(файлово име). От тип nvarchar(32).
* wheel\_diameter – диаметър на колелото. От тип decimal(16,2). Има ограничение да е положително или нула.
* wheel\_width – широчина на колелото. От тип decimal(16,2). Има ограничение да е положително или нула.

Таблицата **Sensors** съдържа следните колони:

* id – уникален идентификатор на сензора. От тип int. Първичен ключ на таблицата. Генерира се автоматично от базата.
* name – име на сензора. От тип nvarchar(128).
* value\_type – тип на отчетената стойност. От тип nvarchar(32).
* sensor\_mesh\_grid – геометрия на сензора(файлово име). От тип nvarchar(32).
* number\_of\_values\_per\_second – брой отчетени стойности за секунда. От тип decimal(8,2). Има ограничение да е положително или нула.

Таблицата **Rotors** съдържа следните колони:

* id – уникален идентификатор на витлото. От тип int. Първичен ключ на таблицата. Генерира се автоматично от базата.
* rotor\_mesh\_grid – геометрия на витлото(файлово име). От тип nvarchar(32).
* rotor\_lifting\_power – подемна мощ на витлото. От тип decimal(16,2). Има ограничение да е положително или нула.

Таблицата **Maps** съдържа следните колони:

* id – уникален идентификатор на картата. От тип int. Първичен ключ на таблицата. Генерира се автоматично от базата.
* map\_data – данни за картата(файлово име). От тип nvarchar(64).
* map\_name – име на картата. От тип nvarchar(64).
* denivelation – максимална денивелация на картата. От тип decimal(16,2).

Таблицата **Algorithms** съдържа следните колони:

* id – уникален идентификатор на алгоритъма. От тип int. Първичен ключ на таблицата. Генерира се автоматично от базата.
* name – име на алгоритъма. От тип nvarchar(64).
* diffEnvironments – дали поддържа различни околни среди като полета, през които да мине. От тип nvarchar(2). Има ограничение да е една от следните две думи- “ДА” или “НЕ”.
* multipleDestPoints – дали поддържа възможност да намери различни крайни точки. От тип nvarchar(2). Има ограничение да е една от следните две думи- “ДА” или “НЕ”.
* complexity – Сложността на алгоритъма. От тип nvarchar(32).
* depth – максималната дълбочина на графа, при която работи алгоритъма.

Таблицата **RoboSimulationsToEnvironments** съдържа следните колони:

* robo\_simulation\_id – уникален идентификатор на симулацията. От тип int. Външен ключ на таблицата. (Таблица RoboSimulations, колона id).
* robo\_ environment\_id – уникален идентификатор на околната среда. От тип int. Външен ключ на таблицата. (Таблица Environments, колона id).

Първичният ключ е наредената двойка (robo\_simulation\_id, environment\_id).

Таблицата **RoboSimulationsToRobots** съдържа следните колони:

* robo\_simulation\_id – уникален идентификатор на симулацията. От тип int. Външен ключ на таблицата. (Таблица RoboSimulations, колона id).
* robot\_id – уникален идентификатор на робота. От тип int. Външен ключ на таблицата. (Таблица Robots, колона id).

Първичният ключ е наредената двойка (robo\_simulation\_id, robot\_id).

Таблицата **RobotsToWheels** съдържа следните колони:

* robot\_id – уникален идентификатор на робота. От тип int. Външен ключ на таблицата. (Таблица Robos, колона id).
* wheel\_id – уникален идентификатор на колелото. От тип int. Външен ключ на таблицата. (Таблица Wheels, колона id).

Първичният ключ е наредената двойка (robot\_id, wheel\_id).

Таблицата **RobotsToSensors** съдържа следните колони:

* robot\_id – уникален идентификатор на робота. От тип int. Външен ключ на таблицата. (Таблица Robos, колона id).
* sensor\_id – уникален идентификатор на сензора. От тип int. Външен ключ на таблицата. (Таблица Sensors, колона id).

Първичният ключ е наредената двойка (robot\_id, sensor\_id).

Таблицата **RobotsToRotors** съдържа следните колони:

* robot\_id – уникален идентификатор на робота. От тип int. Външен ключ на таблицата. (Таблица Robos, колона id).
* rotor\_id – уникален идентификатор на витлото. От тип int. Външен ключ на таблицата. (Таблица Rotors, колона id).

Първичният ключ е наредената двойка (robot\_id, rotor\_id).

Таблицата **RoboSimulationsToMaps** съдържа следните колони:

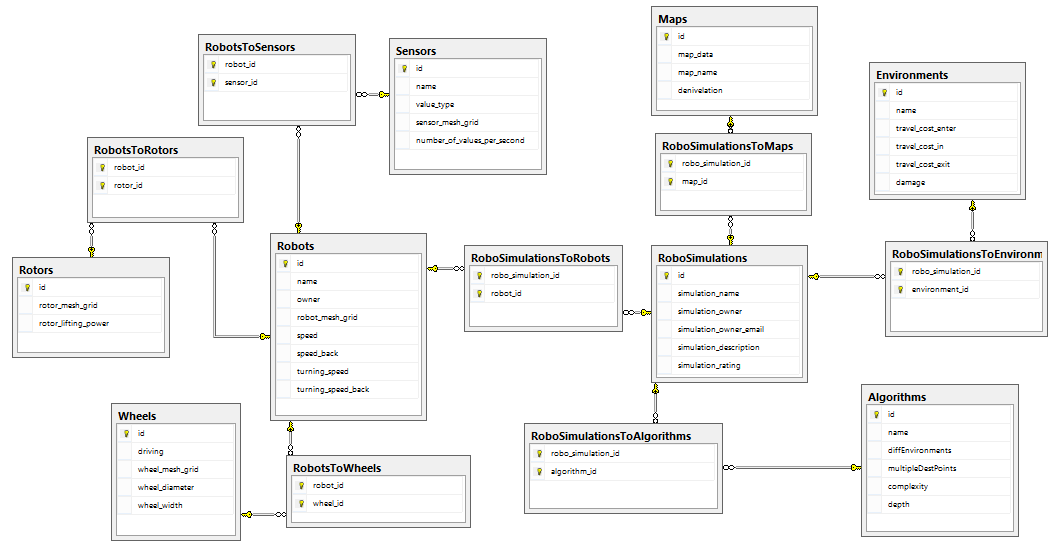
* robo\_simulation\_id – уникален идентификатор на симулацията. От тип int. Външен ключ на таблицата. (Таблица RoboSimulations, колона id).
* map\_id – уникален идентификатор на картата. От тип int. Външен ключ на таблицата. (Таблица Maps, колона id).

Първичният ключ е наредената двойка (robo\_simulation\_id, map\_id).

Таблицата **RoboSimulationsToAlgorithms**съдържа следните колони:

* robo\_simulation\_id – уникален идентификатор на симулацията. От тип int. Външен ключ на таблицата. (Таблица RoboSimulations, колона id).
* algorithm\_id – уникален идентификатор на алгоритъма. От тип int. Външен ключ на таблицата. (Таблица Algorithms, колона id).

Първичният ключ е наредената двойка (robo\_simulation\_id, algorithm\_id).

**7.** **Схема на взаимовръзката между таблиците на БД.**

**8. Прехвърляне на информацията от XML към БД.**

На следващата таблица е показано кои елементи и атрибути от xml документа, в кои таблици и колони в базата от данни преминават.

|  |  |  |
| --- | --- | --- |
| Име на XML елемента/атрибута | Име на таблицата в БД | Име на колоната |
| id (RoboSimulation) | RoboSimulations | id |
| name (RoboSimulation) | RoboSimulations | simulation\_name |
| owner (RoboSimulation) | RoboSimulations | simulation\_owner |
| SimulationOwnerEmail | RoboSimulations | simulation\_owner\_email |
| SimulationDescription | RoboSimulations | simulation\_description |
| SimulationRating | RoboSimulations | simulation\_rating |
| id (Environments) | Environments | id |
| name (Environments) | Environments | name |
| TravelCostEnter | Environments | travel\_cost\_enter |
| TravelCostIn | Environments | travel\_cost\_in |
| TravelCostExit | Environments | travel\_cost\_exit |
| Damage | Environments | damage |
| id (Robots) | Robots | id |
| name (Robots) | Robots | name |
| owner (Robots) | Robots | owner |
| RobotMeshGrid | Robots | robot\_mesh\_grid |
| Speed | Robots | speed |
| SpeedBack | Robots | speed\_back |
| TurningSpeed | Robots | turning\_speed |
| TurningSpeedBack | Robots | turning\_speed\_back |
| driving (Wheel) | Wheels | driving |
| WheelMeshGrid | Wheels | wheel\_mesh\_grid |
| WheelDiameter | Wheels | wheel\_diameter |
| WheelWidth | Wheels | wheel\_width |
| name (Sensor) | Sensors | name |
| valueType (Sensor) | Sensors | value\_type |
| SensorMeshGrid | Sensors | sensor\_mesh\_grid |
| NumberOfValusPerSecond | Sensors | number\_of\_values\_per\_second |
| RotorMeshGrid | Rotors | rotor\_mesh\_grid |
| RotorLiftingPower | Rotors | rotor\_lifting\_power |
| id (Map) | Maps | id |
| name (Map) | Maps | map\_name |
| MapData | Maps | map\_data |
| Denivelation | Maps | denivelation |
| id (Algorithm) | Algorithms | id |
| name (Algorithm) | Algorithms | name |
| diffEnvironments (Algorithm) | Algorithms | diffEnvironments |
| multipleDestPoints (Algorithm) | Algorithms | multipleDestPoints |
| Complexity | Algorithms | complexity |
| Depth | Algorithms | depth |

\* Атрибутите имат в скоби добавено елементът, на който принадлежат!

**9. Приложение, работещо с XML файловете и БД.**

Приложението, което потребителят използва, за да постигне целите на проекта, се състои от две главни функции и набор от вторични, помощни функции:

* Главни функции:
  + „ButtonAddXMLFiles\_Click”
    - описание - функцията има за цел да запише в базата от данни набор от XML файлове, намиращи се на дадено място.
    - тип – void.
    - параметри:
      * „DirPath” (string DirPath = Server.MapPath("~/resources/")) - посочва мястото, от което файловете трябва да се извлекат.
    - използвани помощни функции:
      * „ValidateXMLFIles”
      * „InsertXmlDataToDB”
  + „ButtonAddXMLFiles\_Click”
    - описание - функцията има за цел да създаде нов запис в базата данни с информация, въведена от форма, предложена на потребителя за попълване. Освен това функцията създава и XML файл, съдържащ същата информация.
    - тип – void.
    - параметри:
      * Неограничен брой параметри от тип „TextBox”, отговарящи на атрибутите и поделементите(и техните такива) на елемента „RoboSimulation”. Както и за всеки „TextBox” съответен „Label”, отговарящ за съобщение за невалидни данни в „TextBox-а”.
      * „LabelValidationStatus” - от тип „Label“, в който се записват резултатите от записа в базата от данни и XML документа.
    - използвани помощни функции:
      * „ValidateRoboSimulation”
      * „InsertRoboSimulationToDB”
      * „SaveToXML”
* Вторични помощни функции:
  + „ValidateXMLFIles”
    - описание - функция, която проверява дали XML документите са валидни.
    - тип – void.
    - параметри:
      * „TextBoxValidationStatus” - от тип „TextBox“, в който се записват резултатите от функциите на приложението.
  + „ InsertXmlDataToDB”
    - описание - функция, която въвежда информацията от XML документите в базата от данни.
    - тип – void.
    - параметри:
      * „processor” - от тип „XMLRoboSimulationProcessor“, чрез който се извлича информацията от XML документите.
  + „ ValidateRoboSimulation”
    - описание - функция, която проверява дали всички въведени данни от потребителя във формата са валидни и ако е така- връща true.
    - тип – bool.
    - параметри:
      * „rs” - от тип „RoboSimulation“, чрез който достъпва информацията за полетата.
  + „ InsertRoboSimulationToDB”
    - описание - функция, която въвежда обект от тип „RoboSimulation“ в базата от данни.
    - тип – void.
    - параметри:
      * „ roboSimulation” - от тип „RoboSimulation“, който съдържа информацията от попълнената форма.
  + „ SaveToXML”
    - описание - функция, която въвежда информацията от попълнената форма в XML документ.
    - тип – void.
    - параметри:
      * „rs” - от тип „RoboSimulation“, който съдържа информацията от попълнената форма.
      * „filePath“ (string filePath = Server.MapPath("~/resources/Generated/" + rs.simulation\_name + " " + DateTime.Now.ToString("yyyy-MM-dd HH-mm-ss") + ".xml")) – пълният път с удобно форматирано име на новогенерирания XML документ.