**Задачи к экзамену**

**Задание 1**

1. Разработка ASP.NET HTTP-обработчика. Обработчик обрабатывает GET и POST-запросы. Продемонстрировать с помощью POSTMAN и браузера.

**Задание 2**

1. Разработка ASP.NET MVC-приложения. Реализовать: контроллер с именем FIT, имеющего следующие методы (акции) и представления

|  |  |  |
| --- | --- | --- |
| Акции | Метод | Назначение |
| Index | GET | по умолчанию, не принимает параметров, вызывает представление Index |
| Search | GET | принимает параметр: s(string); вызывает представление Search |
| GetParm | GET | не принимает параметров, вызывает представление GetParm |
| ParmSave | POST | принимает четыре параметра: x(int), s(string), d(date), press (string); вызывает представление ParmSave (клавиша OK) или переадресация на Index (клавиша Cancel) |

|  |  |
| --- | --- |
| Представление | Назначение |
| Index | заголовок INDEX  ссылка на Search с заданным параметром s  ссылка на GetParm |
| Search | заголовок SEARCH  отобразить значение s  ссылка на Index |
| GetParm | заголовок GETPARM  POST-форма для ввода: x, s, d, клавиши (OK, Cancel), переход на ParmSave  ссылка на Index |
| ParmSave | заголовок PARMSAVE  отобразить x, s, d  ссылка на Index |

**Задание 3**

1. Разработка ASMX-приложения. Реализовать ASMX-сервис. Сервис обеспечивает следующие методы.

|  |  |
| --- | --- |
| Метод | Назначение |
| Sum | параметры: x (int), y(int)  результат: x+y |
| Mul | параметры: x (int), y(int)  результат: x\*y |
| Sub | параметры: x (int), y(int)  результат: x-y |

1. Разработать клиент (любое .NET-приложение) демонстрирующее работу сервиса.
2. Продемонстрируйте WSDL-файл.

**Задание 4**

1. Разработка ASP.NET WEB API-приложения. Реализовать REST-сервис. Сервис обеспечивает следующие методы.

|  |  |
| --- | --- |
| Метод | Назначение |
| GET | принимает параметры: s(string)  возвращает JSON-строку:  {  method:GET,  s: *значение s*  } |
| POST | принимает параметры: x(int), s(string), d(date)  возвращает JSON-строку:  {  method: POST,  x:*значение x*,  s: *значение s*,  d: *значение d*  } |
| PUT | принимает параметры: x(int), s(string), d(date)  возвращает JSON-строку:  {  method: PUT,  x:*значение x*,  s: *значение s*,  d: *значение d*  } |
| DELETE | принимает параметры: x(int)  возвращает JSON-строку:  {  method: DELETE,  x:*значение x*,  } |

1. REST-сервис возвращает статус: 200, если все параметры заданы; 400, если не все параметры заданы.
2. Продемонстрировать с помощью POSTMAN

**Задание 5**

1. Разработка ASP.NET COTE MVC-приложения. Реализовать: контроллер с именем FIT, имеющего следующие методы (акции) и представления. Использовать атрибутную форму маршрутизации.

|  |  |  |
| --- | --- | --- |
| Акции | Метод | Назначение |
| Index | GET | по умолчанию, не принимает параметров, вызывает представление Index |
| Search | GET | принимает параметр: s(string); вызывает представление Search |
| GetParm | GET | не принимает параметров, вызывает представление GetParm |
| ParmSave | POST | принимает четыре параметра: x(int), s(string), d(date), press (string); вызывает представление ParmSave (клавиша OK) или переадресация на Index (клавиша Cancel) |

|  |  |
| --- | --- |
| Представление | Назначение |
| Index | заголовок INDEX  ссылка на Search с заданным параметром s  ссылка на GetParm |
| Search | заголовок SEARCH  отобразить значение s  ссылка на Index |
| GetParm | заголовок GETPARM  POST-форма для ввода: x, s, d, клавиши (OK, Cancel), переход на ParmSave  ссылка на Index |
| ParmSave | заголовок PARMSAVE  отобразить x, s, d  ссылка на Index |

1. **КОНЕЦ**