Coursework – Speech

|  |  |
| --- | --- |
| Slide # |  |

1. The topic of my coursework is graph visual editing. So I had to develop an application which is in fact a visual graph editor.
2. There’s the list of requirements on this slide.  
   The application should have the very basic features such as add nodes, remove nodes, move nodes, specify caption for the node, draw links between nodes and so on. The possibility of storing documents in files is an important requirement too.
3. The application is written in C# programming language. So it’s based on the Microsoft .NET framework. I used Microsoft Visual Studio Express 2012 integrated development environment. Since it’s a visual editor, the user interface is an important part of this application, so to build it I used Windows Presentation Foundation framework.
4. I also used git source code management system with a graphical frontend. I stored my git repositories online on GitHub hosting service. And I used NLog logging library to debug my application.
5. Here’s how the application window looks. You see some document being displayed. It contains several nodes and links. All operations are accessible through the main menu and context menus of the nodes.
6. This application provides the very basic functionality. The user can add new nodes to the document, then remove any node. It is possible to rename nodes; draw links between nodes, then again remove those links. The user can scroll large document using the scroll bars.
7. One of the important features of the application is the ability to store graphs inside XML documents so they can be easily accessed, for example, from other applications. These documents have the following format: in the root element named “content” there are NodeControl and LinkControl elements. Each represents a node or a link respectively. NodeControl elements have text and position attributes while each LinkControl element has two attributes which store unique IDs of the nodes which are interconnected by this particular link.
8. While working on this project I got to use Windows Presentation Foundation framework. So I wanted to mention some of its features. It supports hardware acceleration and it is resolution independent.
9. It also supports styles and it’s actually possible to create really good-looking interfaces with it. Like this one. Even though the application I developed is not very complex [kəm'pleks], this framework saved me a lot of time.
10. One of the key features of the Windows Presentation Foundation is that it provides means for separating user interface design from the application source code responsible for application logic. The user interface should be specified in XAML, an xml-like markup language. Here’s an example from my application. This code specifies part of the main window.
11. Windows presentation foundation introduces dependency properties and provides means for connecting properties of different objects like, for example, some data and a window control which displays this data, so when the data changes, the visual representation of this data gets updated immediately. This concept is called Data binding and updating properties is not its only feature. These are pieces of source code from my coursework which are related to dependency properties. Using dependency properties makes the application more complex to some point.
12. There are also two notable features of the C# language which are still relatively new while I believe they are well-known among C# developers. LINQ (читается “link”) is a set of special features for searching lists and processing various data. I tried to use it while working on the project. For example, this expression searches for links connected to the node. Of course, LINK’s usage is not limited by such simple cases. It also has features for mapping .net objects to relational database tables and many others.
13. And another feature is lambda ['læmdə] expression. Which is, in fact, a declaration of an anonymous function which can be constructed inside another function and passed to other functions as an argument so they may call it later when necessary. This code is also taken from my project and it sets dependency property value for every element of the list which matches specified type.
14. This brings me to the end of my presentation.