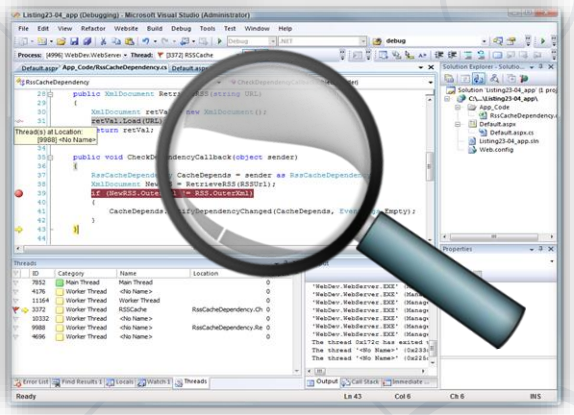
Debugging

Building Rock-Solid Software



SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Table of Contents

Have a Question?

Software University

sli.do

#csharp-advanced

Introduction to Debugging

What is Debugging?

- The process of locating and fixing or bypassing **bugs** (errors) in computer program code
- To **debug** a program:
  - Start with a **problem**
  - Isolate the **source** of the problem
  - **Fix** it
- **Debugging tools** (called **debuggers**) help identify coding errors at various development stages

Debugging vs. Testing

- **Testing**
  - A means of initial detection of errors
  - The process of verifying and validating that a software or application is bug free
- **Debugging**
  - A means of diagnosing and correcting the root causes of errors that have already been detected
  - The process of identifying, analyzing and fixing a bug in the software

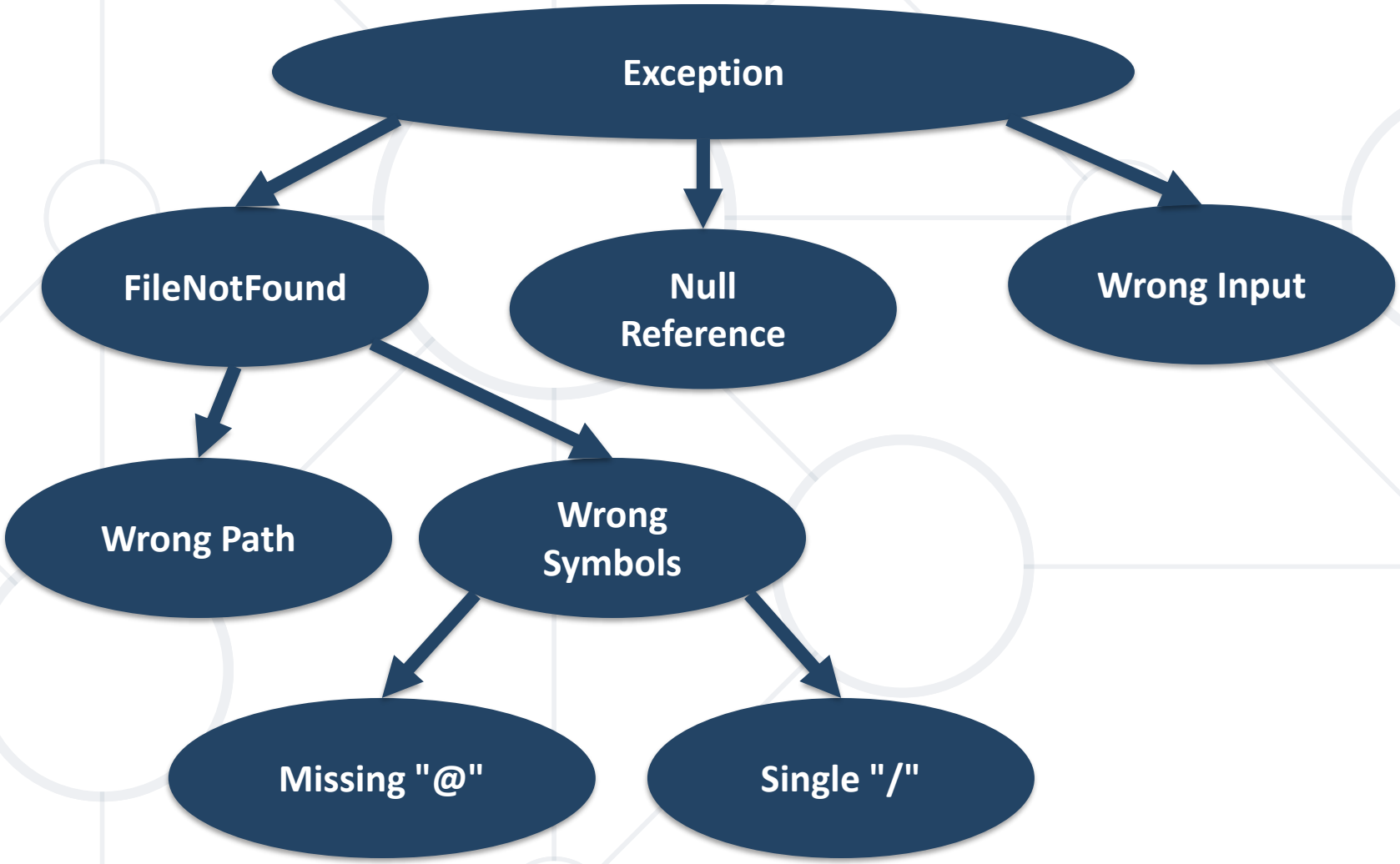
Importance of Debugging

- $60 Billion per year in economic **losses** due to software **defects**
  - E.g. the **Cluster spacecraft failure** was caused by a bug
- Perfect code is an **illusion**
  - There are factors that are out of our control
- **Legacy** code
  - You should be able to debug code that is written years ago
- **Deeper understanding** of system as a whole

Debugging Philosophy

- Debugging can be viewed as one big **decision tree**
  - Individual nodes represent **theories**
  - Leaf nodes represent possible **root causes**
  - Traversal of tree boils down to process state **inspection**
  - Minimizing time to resolution is **key**
    - Careful traversal of the decision tree
    - Pattern recognition
    - Visualization and ease of use helps minimize time to resolution

Example Debugging – Decision Tree



Visual Studio Debugger

Visual Studio Debugger

Software University

- Visual Studio IDE gives us a lot of **tools** to **debug** your application
  - Adding **breakpoints**
  - Visualize the **program flow**
  - Control the **flow of execution**
  - **Data tips**
  - **Watch variables**
  - Debugging **multithreaded programs**
  - And many more…

Debugging a Solution

- Debug menu, Start Debugging item
  - **F5** is a shortcut
- Easier access to the source code and symbols since its loaded in the solution
- Certain differences exist in comparison to debugging an already running process
  - Hosting for an **ASP.NET application**
    - Visual Studio uses a replacement of the real IIS

Debug Windows

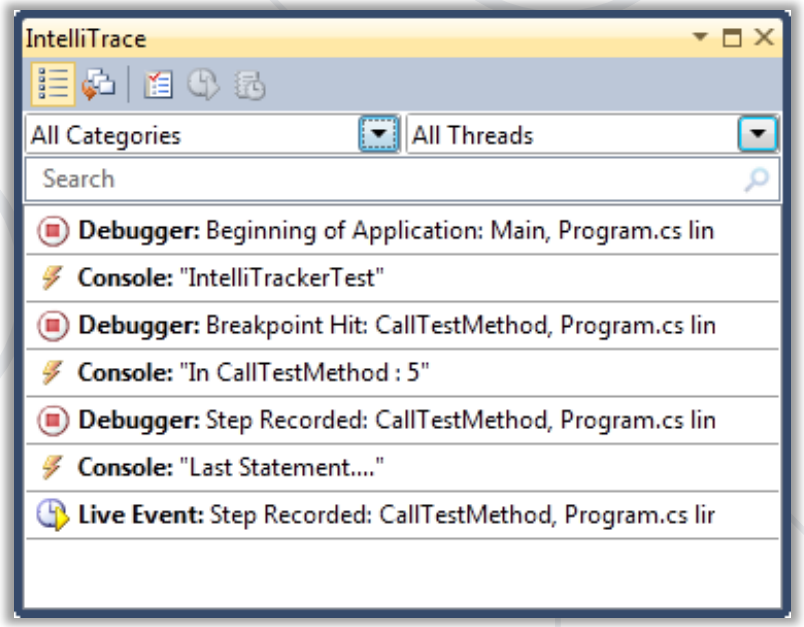
- Debug Windows are the means to introspect on the state of a process
- Opens a new window with the selected information in it
- Window categories
  - **Data inspection**
  - **Threading**
- Accessible from menu
  - **Debug** -> **Windows**

Debugging Toolbar

- Convenient shortcut to common debugging tasks
  - **Step into**
  - **Step over**
  - **Continue**
  - **Break**
  - **Breakpoints**
- Customizable to fit your needs
  - **Add** / **Remove** buttons

IntelliTrace

- **IntelliTrace** operates in the background, records what you are doing during debugging
- You can easily get a past state of your application from IntelliTrace
- You can **navigate through** your code and see what's happened
- To navigate, just click any of the events that you want to explore



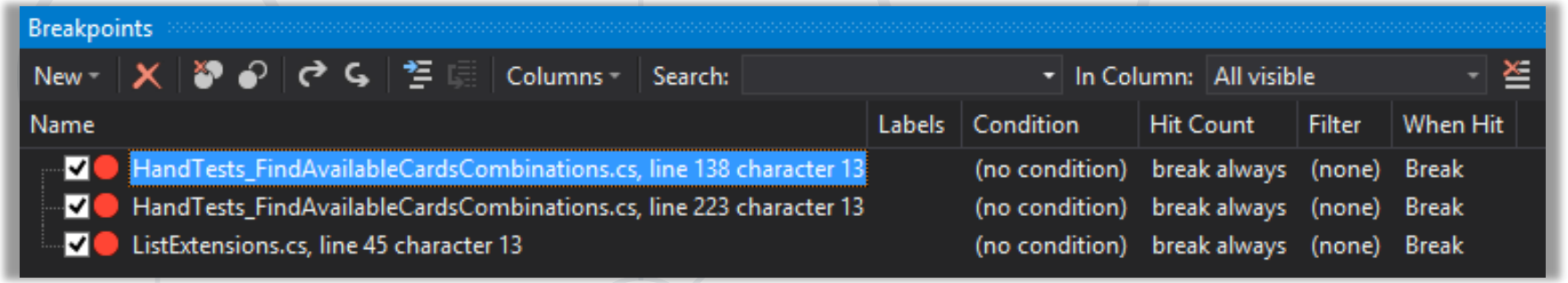
Breakpoints

Breakpoints

- Ability to stop execution based on certain criteria is key when debugging
  - When a **function is hit**
  - When **data changes**
  - When a specific **thread** hits a **function**
  - Much more…
- Visual Studio's debugger has a huge feature set when it comes to breakpoints

Managing Breakpoints

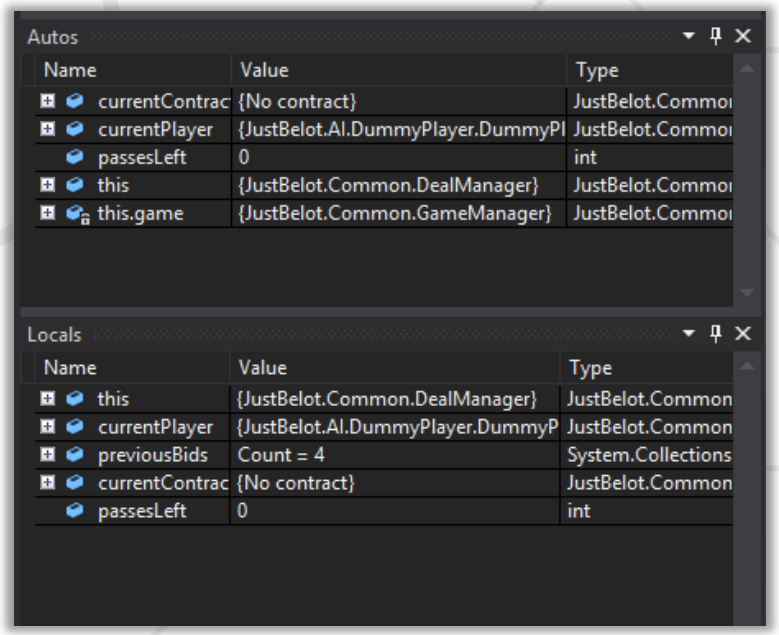
- Managed in the breakpoint window
- Adding breakpoints
- Removing or **disabling** breakpoints
- **Labeling** or **grouping** breakpoints
- Export/import breakpoints



Data Inspection

Visual Studio Data Inspection

- Visual Studio offers great data inspection features
  - **Watch windows**
  - **Autos** and **Locals**
  - **Memory** and **Registers**
  - **Data Tips**
  - **Immediate** window



Watch Window

- Allows you to inspect various states of your application
- Several different kinds of "**predefined**" watch windows
  - **Autos**
  - **Locals**
- "**Custom**" watch windows also possible
  - Contains only variables that you choose to add
  - Right click on the variable and select "Add to Watch"

Autos and Locals

- Locals watch window contains the local variables for the specific stack frame
  - **Debug** -> **Windows** -> **Locals**
  - Displays: name of the variable, value and type
  - Allows drill down into objects by clicking on the + sign in the tree control
- **Autos** lets the debugger decide which variables to show in the window
  - Loosely based on the current and previous statement

Memory and Registers

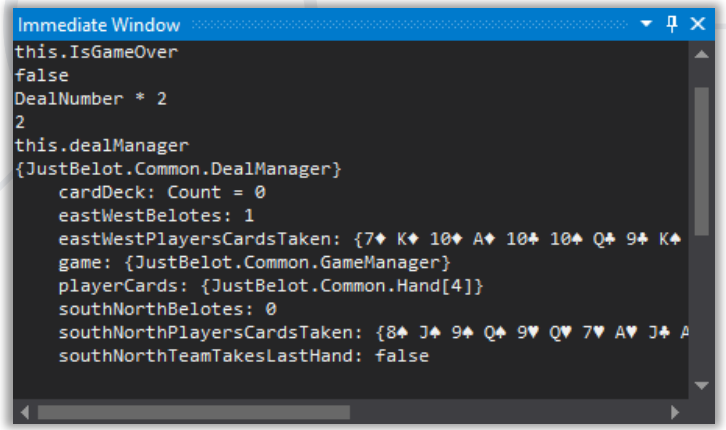
- **Memory** window can be used to inspect process wide memory
  - Address field can be a raw pointer or an expression
  - Drag and drop a variable from the source window
  - Number of columns displayed can be configured
  - Data format can be configured
- **Registers** window can be used to inspect processor registers

Data Tips

- Provides information about variables
  - Variables must be within scope of current execution
- Place mouse pointer over any variable
  - Variables can be expanded by using the + sign
- Pinning the data tip causes it to always stay open
- Comments can be added to data tips
- Data tips support drag and drop
- Importing and exporting data tips

Immediate Window

- Useful when debugging due to the **expansive expressions** that can be **executed**
  - To output the value of a variable {**name of variable**}
  - To set values, use {**name of variable**}={**value**}
  - To call a method, use {**name of variable**}.<**method**>(**arguments**)
  - Similar to regular code
  - Supports **IntelliSense**



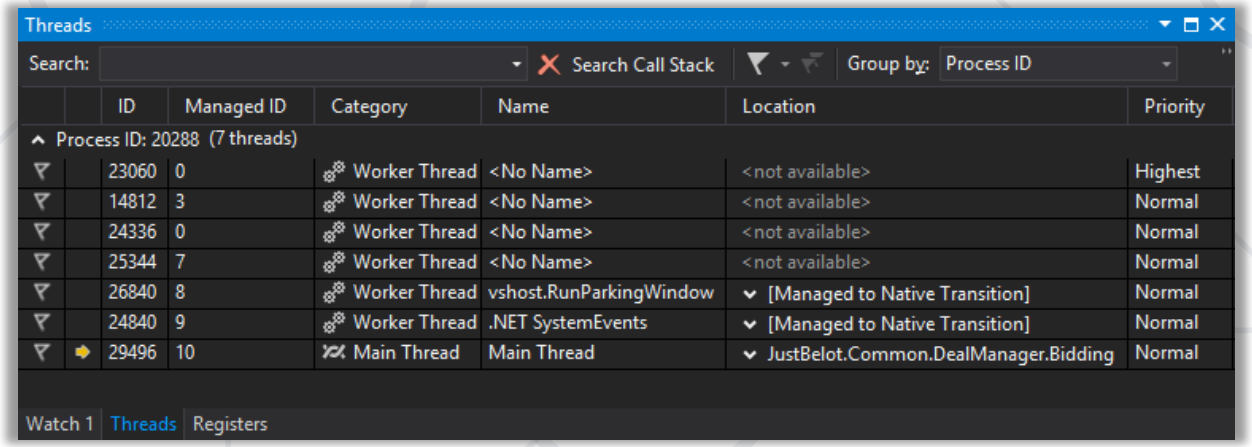
Threads and Stacks

Threads

- Fundamental units of code execution
- Commonly, programs use **more** than **one thread**
  - In .NET, there is always more than one thread
- Each thread has a memory area associated with it known as a **stack**
  - Stores **local variables**
  - Stores frame **specific information**
- Memory area employs last in first out semantics

Threads Window

- Contains an overview of thread activity in the process
- Includes basic information in a per thread basis
  - Thread ID's
  - Category
  - Name
  - Location
  - Priority

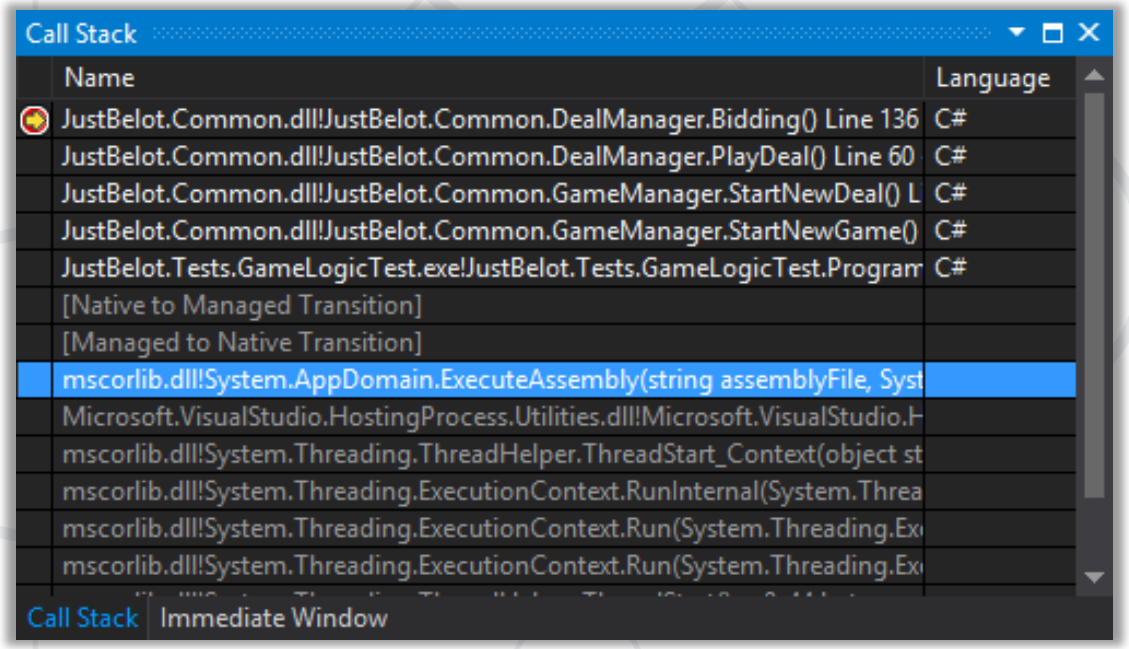


Breakpoint Filters

- Allows you to excerpt even more control of when a breakpoint hits
- Examples of customization
  - Machine **name**
  - Process **ID**
  - Process **name**
  - Thread **ID**
  - Thread **name**
- Multiple can be combined using &, ||, !

Call Stacks

- Visual Studio shows the elements of a call stack
  - Local variables
  - Method frames



Finding a Defect

Tips for Finding Defects (1)

- Use all available data
- Refine the test cases
- Check unit tests
- Use available tools
- Reproduce the error in several different ways
- Generate more data to generate more hypotheses
- Use the results of negative tests
- Brainstorm for possible hypotheses

Tips for Finding Defects (2)

- Narrow the suspicious region of the code
- Be suspicious of classes and routines that have had defects before
- Check code that's changed recently
- Expand the suspicious region of the code
- Integrate incrementally
- Check for common defects
- Talk to someone else about the problem
- Take a break from the problem

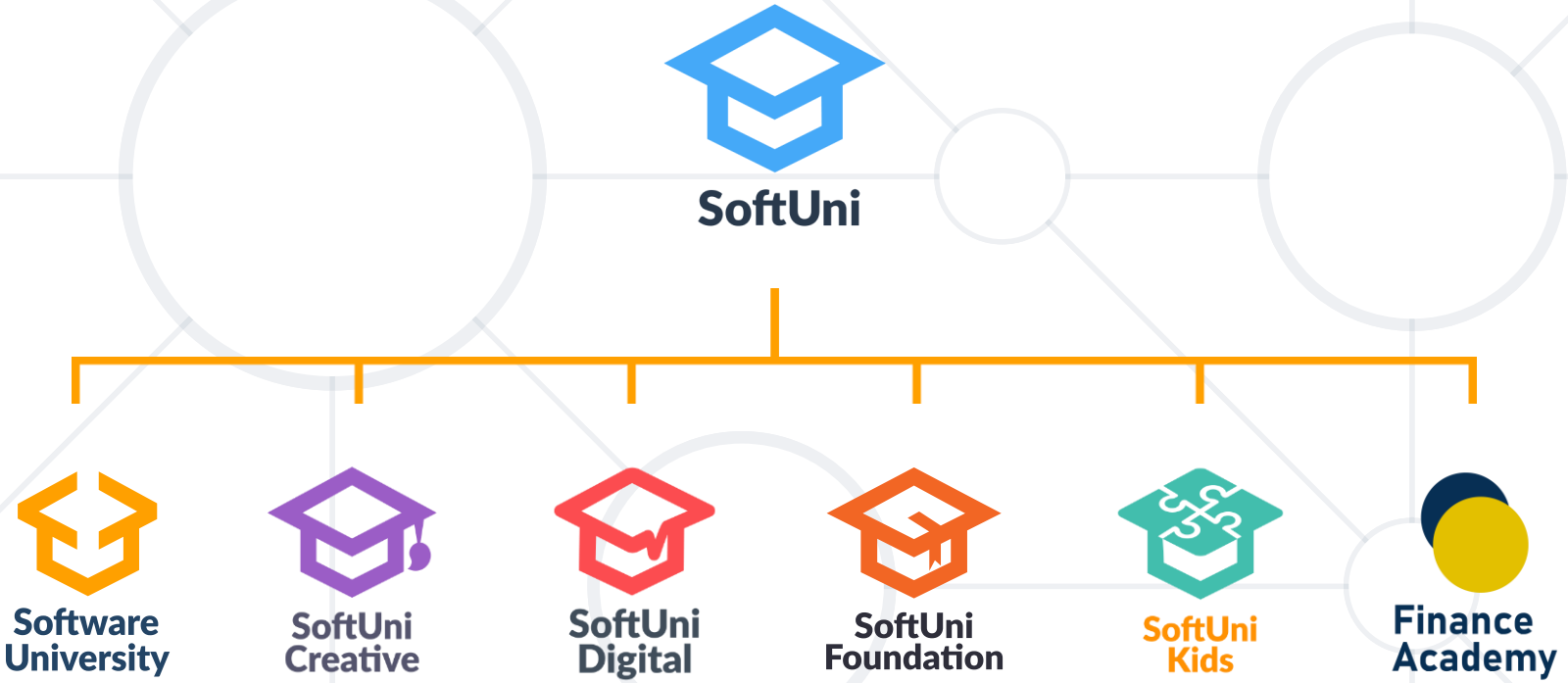
Fixing a Defect

- Understand the problem before you fix it
- Understand the program, not just the problem
- Confirm the defect diagnosis
- Relax
- Save the original source code
- Fix the problem, not the symptom
- Make one change at a time
- Add a unit test that expose the defect
- Look for similar defects

Summary

- Introduction to **Debugging**
- Visual Studio Debugger
- **Breakpoints**
- Data Inspection
  - **Locals**, **Autos**, **Watch**
- Finding a **Defect**

Questions?



SoftUni Diamond Partners

Software University

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg
- © Software University – https://softuni.bg