Техники за дебъгване

Дебъгване – процесът на намиране и оправяне на даден проблем/бъг

За да идентифицираме бъга по-лесно, трябва да изолираме кода колкото се може повече

**Perfect code is an illusion!**

Писането на тестове ни помага да намерим бъговете преди тяхната поява

Чрез дебъгване търсим/намираме вече появили се бъгове

Когато дебъгваме даден код работим на принципа - последователно изключване на възможните грешки

Понякога когато попаднем на стар код и в него има даден бъг, то на този бъг може да се разчита от друг код, така че дори да оправим дадения бъг, не е гаранция, че ще сме оправили проблема като цяло

Watch прозореца (VS) – в него може да си вкарваме променливи, чиито стойности може да следим през цялото време на дебъгване

Immediate window – начин да променяме стойностите на дадени променливи в runtime

Breakpoints window – доста полезен когато имаме много класове

IntelliTrace прозореца пази история за редовете, през които е минал дебъгера

Fixing a defect:

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

Всеки един thread (нишка) има собствен call stack, в който се пазят методите, които се извикват, по време на изпълнението на конкретната нишка

Нормално е голяма част от времето в работата да минава в дебъгване

Без дебъгване, грешките се откриват много трудно