Защитно програмиране

Defensive programming – to always expect incorrect input and problem situations and handle them correctly

Всички входни данни, които влизат в нашата програма, трябва да бъдат валидирани

При добрият софтуер данните се валидират и в data layer- а, и в клиента, и в бизнес логиката

Начини за поведение при проблемни ситуации:

* Връщане на неутрална стойност - например при JavaScript ако искаме да парснем стринг с метода parseInt, то ни се връща NaN, или -1 при IndexOf() в C#
* Да върнем някакви валидни данни – например при “Hi”.substr(5, -3) при JavaScript се връща празен стринг “”
* Хвърляне на exception- и – типично за обектно-ориентираните езици
* Показване на error message в UI- а – например при въвеждане на невалидни данни в input поле
* Log a message and stop the script, e.g. die() in PHP – за фатални грешки
* Return the same answer as the previous time
* Crash / shutdown / reboot – при крайност

Assert – means to guess

Debug.Assert() – се използва при писане на кода за помощ на програмиста

Assertion – ите трябва да се използват само когато се появяват фатални грешки

Avoid putting executable code in assertions:

Wrong:

Debug.Assert(PerformAction(), “Could not perform action”);

Correct:

bool actionPerformerd = PerformAction();

Debug.Assert(actionPerformed, “Could not perform action”);

**Assertion == fatal error!**

Трябва да хвърляме exception- и само в случаи, които са настина грешни и некоректни, а при по- малки грешки като например грешно въведена парола, може да кажем на user-а да въведе отново

Разлика между exception-и и assertion-и:

Exception-ите може да се обработват и те се появяват в release mode.

Assertion-ите не се появяват в release mode и не могат да се обработват

Exception- ите се пишат от програмисти за програмисти, не за потребителите

Съществува chain- ване на exception- и. Т.е. една грешка може да е предизвикана от втора грешка, втората грешка да е предизвикана от трета грешка и т.н.

нашия код никога не трябва да връща грешен резултат!

**Като заключение: няма един единствен правилен начин за справяне с конкретна грешка. При различни ситуации има различни правилни начини за обаботване на грешки.**