

Code Contracts

Deák László

Runtime analysis

Compile time analysis

Unit Testing & Documentation

Why .NET?

Preconditions

- Entering
- No side effect
- Private members
- If-then-throw
- ContractRequires method
- Runtime behaviour is determined by runtime analyzer

Postconditions

- Before exiting
- Runtime behaviour is determined by runtime analyzer
- May reference private members
- Ensures method must be true
- Exceptional Postconditions

Postconditions

- `Contract.Result<T>()`
- `Contract. oldValue<T>(e)`
- Megkötések
- `Contract.ForAll`
- `Contract.Exists`

Object Invariants

- Express conditions under object is correct

```
[ContractInvariantMethod]
```

```
protected void ObjectInvariant ()
```

```
{
```

```
    Contract.Invariant ( this.y >= 0 );
```

```
    Contract.Invariant ( this.x > this.y );
```

```
}
```

Ordering

If-then-throw statements	Backward-compatible public preconditions
Requires	All public preconditions.
Ensures	All public (normal) postconditions.
EnsuresOnThrow	All public exceptional postconditions.
Ensures	All private/internal (normal) postconditions.
EnsuresOnThrow	All private/internal exceptional postconditions.
EndContractBlock	If using if-then-throw style preconditions without any other contracts, place a call to EndContractBlock to indicate that all previous if checks are preconditions.

Purity