

# Program szkolenia:

## Zarządzanie pamięcią w .NET

### Informacje:

<b>Nazwa:</b>	<b>Zarządzanie pamięcią w .NET</b>
<b>Kod:</b>	<b>NETplatform-mem</b>
<b>Kategoria:</b>	Platforma .NET
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

---

Warsztat ma za zadanie zaznajomić uczestników z tematem świadomego programowania i projektowania oprogramowania .NET pod kątem zarządzania pamięcią i wynikającą z niej skalowalnością, wydajnością i kosztami.

Warsztat składa się z trzech nierozłącznych części, przeplatanych ze sobą w trakcie kursu:

#### Programowanie

Dowiedz się jak pisać wydajny kod, uwzględniając różne "humory" Garbage Collectora, o których być może nawet jeszcze nie wiesz. Po tym szkoleniu zaczniesz programować znacznie bardziej świadomie, nie unikając również bardziej zaawansowanych technik.

Np. czy wiesz, że kolejność pól w obiektach może wpływać na wydajność kodu? Czy jesteś świadom ukrytych źródeł alokacji?

#### Teoria

Dowiedz się jak to wszystko działa pod spodem. Wiedza ta nie tylko poszerzenie inżynierskich horyzontów, ale również możliwość lepszego rozumienia i analizowania na własną rękę wszystkiego, na co natkniesz się w przyszłości w temacie zarządzania pamięcią.

Np. czy potrafisz obecnie ze zrozumieniem porównywać GC z różnych ekosystemów? Czy dobrze rozumiesz ogłaszane zmiany w CLR w tym zakresie?

#### Diagnostyka

Naucz się jak diagnozować i monitorować swoje aplikacje. Po zdobyciu tej wiedzy żadne problemy nie będą Ci straszne, nawet "na produkcji". Po tym szkoleniu, rozwiązanie problemów to tylko kwestia czasu. Pytanie nie będzie już brzmieć "jak?" i "czy?" je rozwiążesz, ale jedynie "kiedy?".

Np. czy wiesz jak bezinwazyjnie diagnozować wyciek pamięci albo wysokie użycie CPU przez GC?

### Wymagania

Uczestnicy znają podstawy programowania w języku C# oraz potrafią obsługiwać środowisko programistyczne Visual Studio.

## Zalety szkolenia:

- Przeprowadzone przez autora popularnej książki Pro .NET Memory Management
- Solidna dawka praktyki związanej z programowaniem i diagnostyką, wraz z porządną dawką wiedzy teoretycznej

## Szczegółowy program:

### 1. Podstawowe pojęcia

- 1.1. ręczne i automatyczne zarządzanie pamięcią
- 1.2. system operacyjny, pamięć wirtualna i proces .NET

### 2. Partycjonowanie pamięci

- 2.1. generacje - jaki stan jest zdrowy?
- 2.2. Small and Large Object Heap - co musimy wiedzieć?
- 2.3. segmenty - czy powinniśmy się przejmować?

### 3. Alokacje

- 3.1. jak alokacja działa w SOH i LOH - czy naprawdę jest tania?
- 3.2. najczęstsze źródła alokacji i ich unikanie
- 3.3. stackalloc - poznaj super tani alokator

### 4. Odśmiecanie pamięci

- 4.1. jak to działa - fazy Mark, Plan, Sweep i Compact
- 4.2. OutOfMemoryException - kiedy, dlaczego i jak diagnozować?
- 4.3. wyciek pamięci
- 4.4. tryby Server vs Workstation - który wybrać?
- 4.5. tryby Background vs Non-concurrent - który wybrać?
- 4.6. kontrola GC z kodu

### 5. Czas życia obiektów

- 5.1. explicit cleanup - interfejs IDisposable i tzw. "Disposable pattern"
- 5.2. implicit cleanup - finalizacja
- 5.3. weak references

### 6. Zaawansowane tematy

6.1. używanie struktur - po co?

6.2. ref structs - szczególny przypadek struktur

6.3. Span i Memory - nowi przyjaciele wydajności

6.4. klasa Unsafe i jej potężne możliwości

6.5. Pipelines - nowy nabytek w .NET

## 7. Data-oriented design

7.1. projektowanie aplikacji pod kątem dostępu do pamięci

7.2. metody taktyczne (lokalne)

7.3. metody strategiczne (globalne)

## 8. Automatyzacja diagnostyki

8.1. biblioteka ClrMD - do potężnej diagnostyki

8.2. biblioteka TraceEvent - do potężnego monitoringu (w tym Azure)