



# 3. Архитектура на .NET Framework



# Съдържание

- Какво е .NET Framework?
- Структура на .NET Framework
- Common Language Runtime
- Управляван и неуправляван код
- Модел на компилиране и изпълнение
- .NET програми
- Стандартизация на .NET Framework



# Какво е .NET Framework?

- Платформа за разработка на софтуер
- Среда за изпълнение на програми



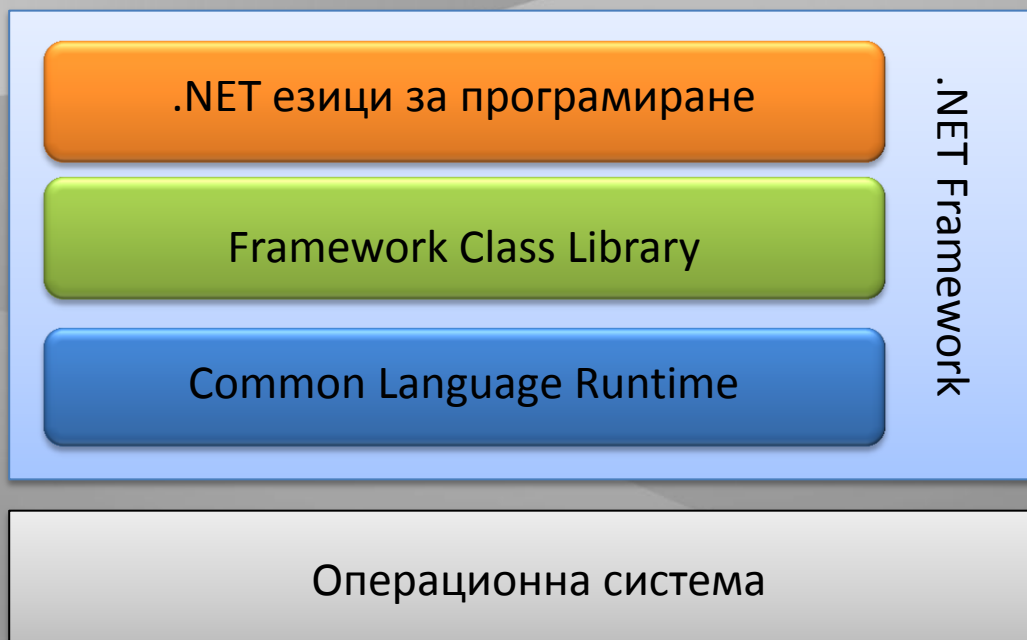
# Характеристики на .NET Framework

- Обща среда за контролирано изпълнение на приложения
- Библиотека с базова функционалност
- Езикова независимост
- Взаимодействие с неуправляван код (Interoperability)
- Преносимост на приложенията



# Компоненти на .NET Framework

- Common Language Runtime (CLR)
- Framework Class Library (FCL)





# Common Language Runtime

- Виртуална машина
- Среда за контролирано изпълнение на управляван код
- Предоставя достъп до ресурсите на ОС



# Компоненти на CLR

## Common Language Runtime

### Base Class Library Support

Thread Support

COM Marshaler

Type Checker

Exception Manager

Security Engine

Debug Engine

JIT Compiler

Code Manager

Garbage Collector

Class Loader



# Отговорности на CLR (1/3)

- Изпълнение на управляван код
  - Intermediate Language (IL)
  - Just-In-Time (JIT) компилатор компилира IL до машинен код
  - Изпълнение на машинния код
- Управление на паметта и ресурсите
  - Грижи се за заделяне на паметта
  - Грижи се за освобождаване на паметта - Garbage Collector





# Отговорности на CLR (2/3)

- **Безопасност на типовете**
  - **Не позволява** директен достъп до паметта
  - **Не позволява** директна работа с указатели
  - **Не позволява** преобразуване м/у несъвместими типове
  - **Не позволява** излизане границите на масив
  - **Не позволява** ползването на неинициализирани променливи



# Отговорности на CLR (3/3)

- Управление на изключенията
- Управление на сигурността
- Управление на конкуретността
- Взаимодействие с неуправляван код
- Подпомага дебъгването и оптимизирането на кода



# Неуправляван код

- Директни инструкции към процесора
- Машиннозависим
- Директен достъп до паметта



# Управляван код

- Инструкции към виртуална машина
- Машиннонезависим
- Контролиран достъп до паметта
- Обектно-ориентиран
- Метаданни описващи кода



# Модел на компилация



- Структура на асембли
  - Метаданни
  - Intermediate Language код
  - Ресурси (икони, текст и др.)
- Асемблото е изпълним файл или динамична библиотека с типове



# Метаданни (1/3)

- Описание на асемблито
  - Име, версия, локализация
  - Цифров подпис
  - Зависимост от други асемблита
  - Др.



# Метаданни (2/3)

- Описание на типовете
  - Класове, интерфейси, типове
  - Базови класове и имплементирани интерфейси
  - Член-променливи и свойства
  - Методи, параметри на методи, връщани стойности
  - Атрибути приложени към класове, методи ...



# Метаданни (3/3)

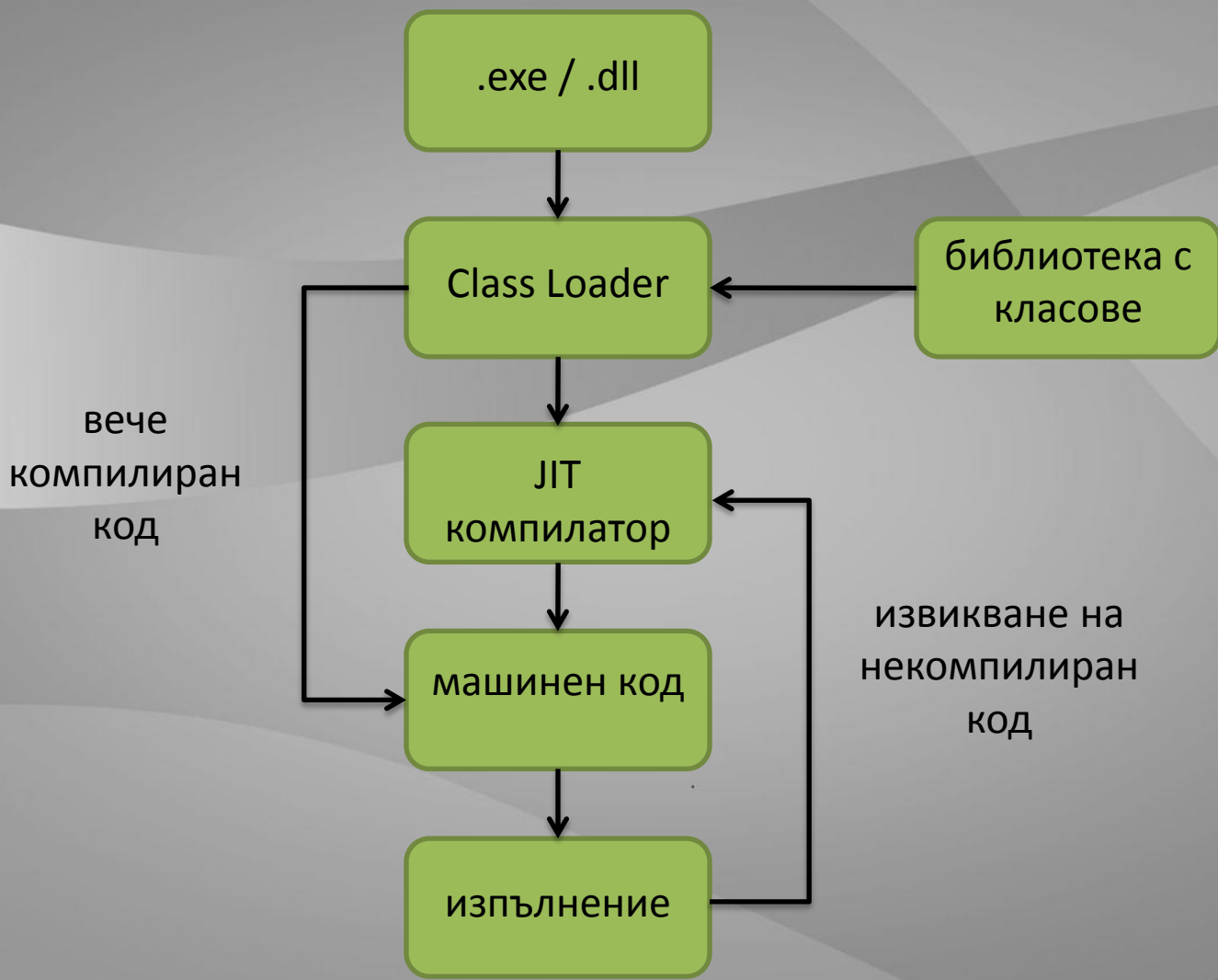
- Приложение на метаданните
  - Реализират типовата безопасност
  - Нужни са на Garbage Collector-а по при освобождаване на ресурси
  - Премахват нуждата от хедър файлове
  - Изполват се от IntelliSense-а





# Модел на изпълнение

- JIT компилация
  - Изпълнява се само веднъж
  - Компилира се само кода който ще се изпълни
  - Оптимизира кода за текущата платформа
  - Прекомпилира се при изпълнение на друга хардуерна или софтуерна платформа
  - Има възможност за предварителна компилация с [ngen.exe](#)





# Managed vs. unmanaged code

- Бързина на изпълнение
- Авторски права
- Оптимизации



# Асембли

- Асемблито е минималната, самостоятелна, градивна част на .NET приложение
- Частни асемблита
- Публични асемблита
- Global Assembly Cache (GAC)
- Assembly Strong Name



# .NET приложение

- Конфигурируеми изпълними единици
- Състоят се от едно или няколко асемблита
- Инсталират се с просто копиране, без сложна регистрация на компонентите



# Application Domains

- Изолирано изпълнение на няколко приложения в рамките на един процес
- Всяко приложение е в отделен App Domain
- Не се нарушава сигурността по време на изпълнение
- CLR процеса не се зарежда за всяка програма
- Повишава се производителността



# Common Type System

- Осигурява съвместимост на типовете между различните .NET езици
- Дефинира поддържаните от CLR типове данни и операциите над тях



# Framework Class Library

- Вградена библиотека с богата функционалност
- Средства за разработване на:
  - Windows клиентски приложения
  - Уеб приложения
  - Конзолни приложения
  - Уеб услуги
  - Windows услуги
  - Други библиотеки с компоненти





# Стандартизация на .NET

- Common Language Infrastructure (CLI):
  - Common Language Specification (CLS)
  - Common Type System (CTS)
  - Common Intermediate Language (CIL)
  - Управление на изключения
  - Асемблита и метаданни
  - Части от Framework Class Library



# Стандартизация на .NET

- Интеграция на различни езици за програмиране
- Междуплатформена съвместимост
  - .NET Framework – имплментаци на CLI за Windows
  - [Mono](#) – имплементация на CLI за Linux и Mac



# Развитие на .NET Framework





Въпроси?



# Ресурси

- “Програмиране за .NET Framework”, Светлин Наков и колектив, Глава 1: Архитектура на .NET Framework
- “CLR via C#”, Jeffrey Richter, Chapter 1: The CLR’s Execution Model
- [“.NET Framework”](#), Wikipedia