Технологии программирования

Лекция №1

ИС, весна 2022

О чём будет курс?

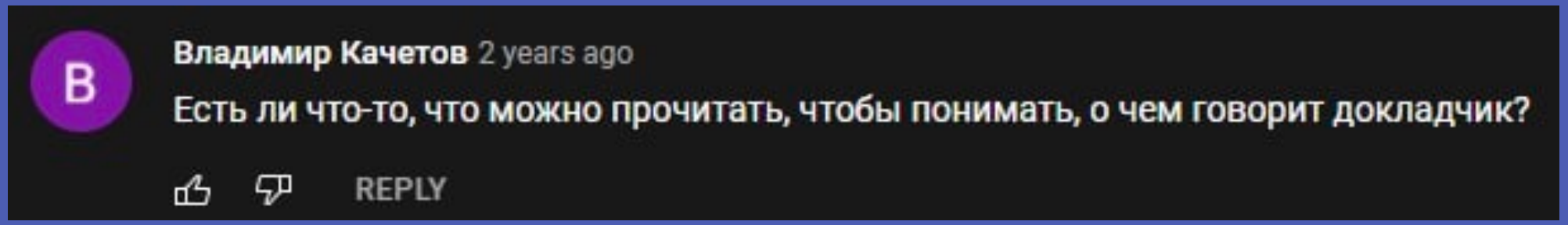
Темы лекций

1. Обзор .NET (и сравнение с JVM)
2. Roslyn: компиляция, анализаторы кода, кодогенерация
3. Runtime, особенности работы, виды
4. Just-in-Time компиляция
5. Работа с памятью, аллокация памяти
6. Garbage collector, особенности, режимы работы, перфоманс
7. Бенчмаркинг, оптимизации

Разбалловка

Работа	Кол-во баллов	Срок сдачи
4-5 лабораторных работ	59	2-3 недели на работу
N тестов	21	на лекциях
Экзамен	20	на сессии
Итого	100	в мечтах
	103	во влажных мечтах Миши Либченко

Литература


B
Владимир Качетов 2 years ago
Есть ли что-то, что можно прочитать, чтобы понимать, о чем говорит докладчик?
REPLY

Литература

- Chris Farrell "Under the Hood of .NET Memory Management"
- Конрад Кокоса Управление памятью в .NET для профессионалов
- Alessandro Del Sole "Roslyn Succinctly"
- Станислав Сидристый ".NET Platform Architecture book"
- Бен Уотсон "Высокопроизводительный код на платформе .NET"
- Эндрю Троелсен "Язык программирования C# 9 и платформа .NET 5"

Исходники dotnet: https://github.com/dotnet

Фаст обзор .NET

Языки в .NET

- Visual Basic
- C++/CLI
- F#
- C#

Языки в .NET: Visual Basic

- RIP?
- VBA
- Один компилятор с C#

Языки в .NET: C++/CLI

- Интероп

Языки в .NET: F# (Scala из JVM)

Донор фичей:

- Иммутабельность
- Records
- Pattern matching
- Expressions вместо Statement

Языки в .NET: C#

Он есть. И он пытается стать F#

- C# 10
- .NET 6

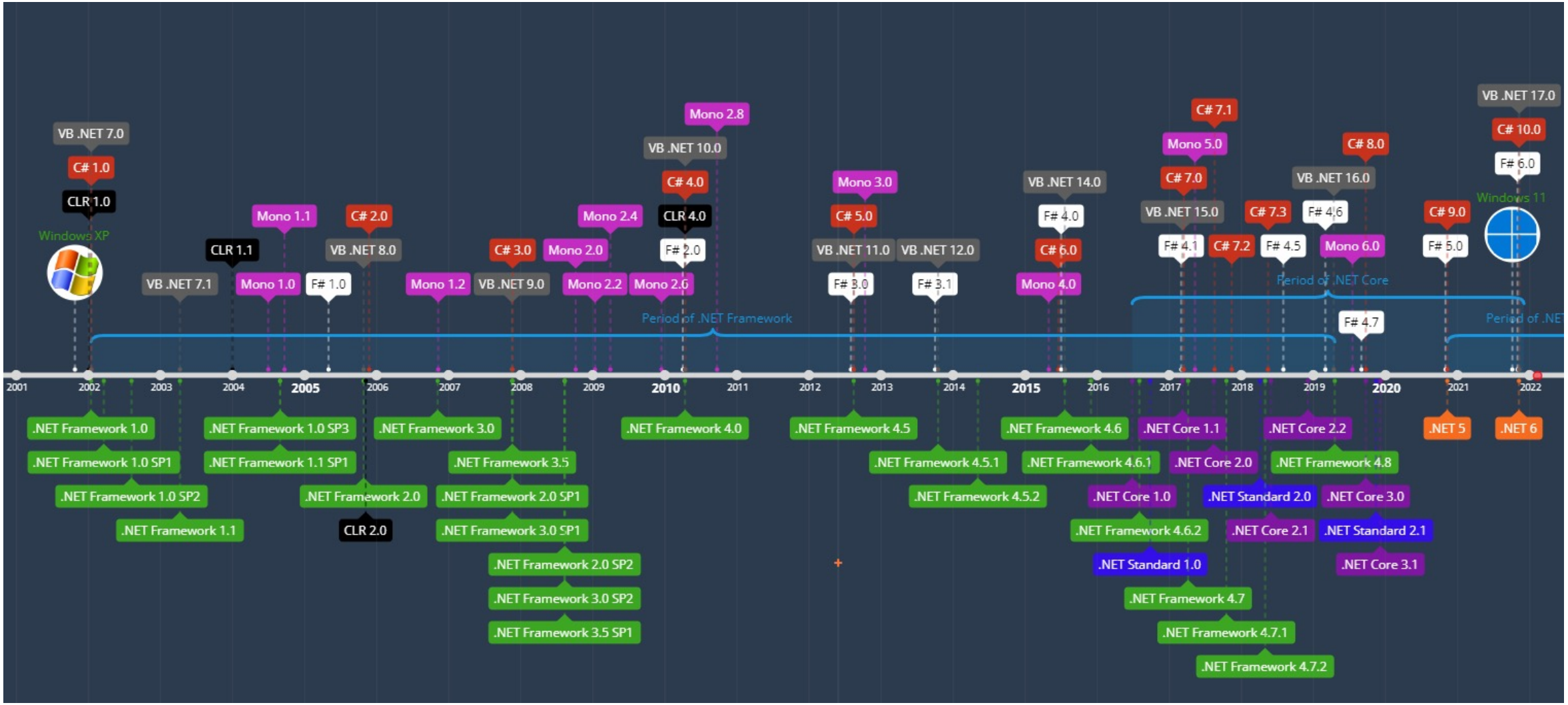
Смешивание языков

- .NET поддерживает мультиязычность благодаря CLR (Common Language Runtime).
- CLR генерирует IL-код, который не привязан к ЯП и впоследствии может быть преобразован в машинный код.
- В рамках одного солюшена можно использовать разные языки, вызывать их друг из друга (с некоторыми ограничениями).
- В отличие от JVM, смешивать языки в .NET можно только попроектово, а не пофайлово. Это связано с тем, что единица компиляции в JVM – файл, а в .NET – проект.

Runtime

- Common language runtime, CLR - рантайм, всё то, что нужно для выполнения кода.
- https://github.com/dotnet/runtime

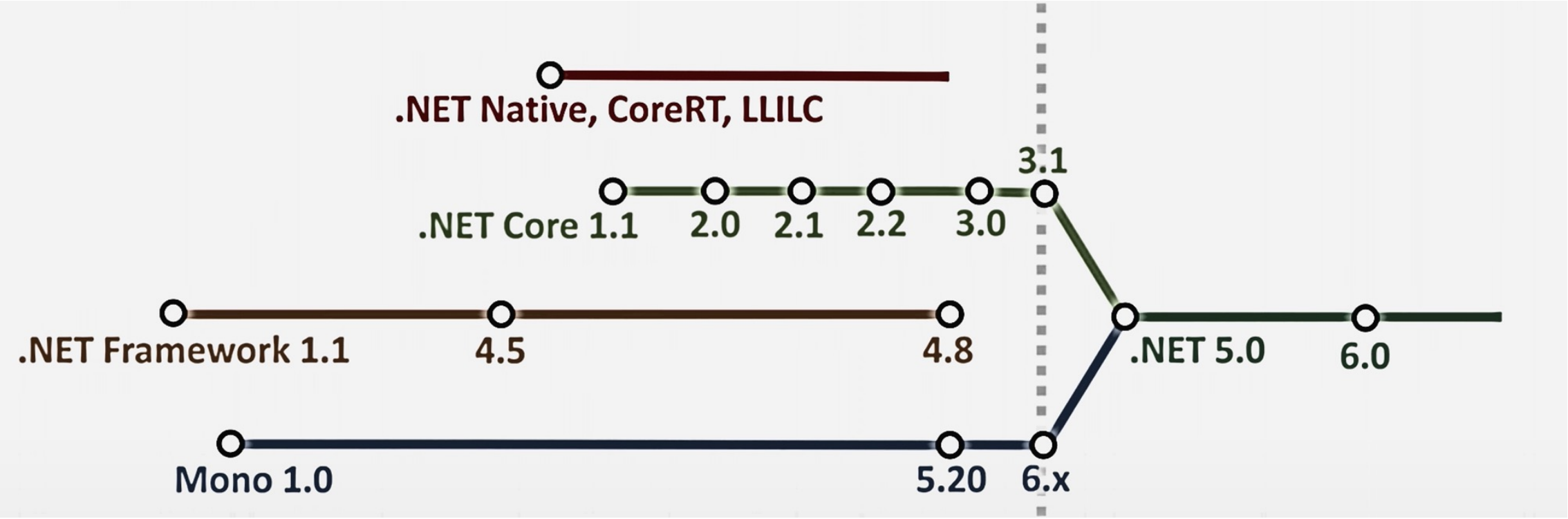
Runtimes: Framework, Core, Mono, .NET 5



Источник: https://time.graphics/line/593132?utm_content=190567820

Runtimes

- .NET framework
- .NET Core
- .NET 5/6
- Mono



IL-код в .NET (= байт код JVM)

- IL (он же иногда MSIL, CIL) - промежуточный язык платформы dotnet

IL-код, C#

```
using static System.Console;
using static System.Linq.Enumerable;


int Square(int x) => x * x;

int SumOfSquares(int n) =>
    Range(1, n)
    .Select(Square)
    .Sum();


WriteLine(SumOfSquares(5));
```

IL-код, C#

```
.method private hidebysig static
    void '<Main>$' (
        string[] args
    ) cil managed
{
    // Method begins at RVA 0x2050
    // Code size 12 (0xc)
    .maxstack 8
    .entrypoint

    IL_0000: ldc.i4.5
    IL_0001: call int32 Program::'<<Main>$>g__SumOfSquares|0_1'(int32)
    IL_0006: call void [System.Console]System.Console::WriteLine(int32)
    IL_000b: ret
} // end of method Program::'<Main>$'
```

Осторожно, фарш!

Убедительная просьба – убрать от экранов детей, беременных женщин и людей с тонкой душевной организацией.

IL-код, F#

```fsharp
let square x = x * x

let sumOfSquares n =
    [1..n]
    |> Seq.map square
    |> Seq.sum

printfn "%d" (sumOfSquares 5)
```

IL-код, F#

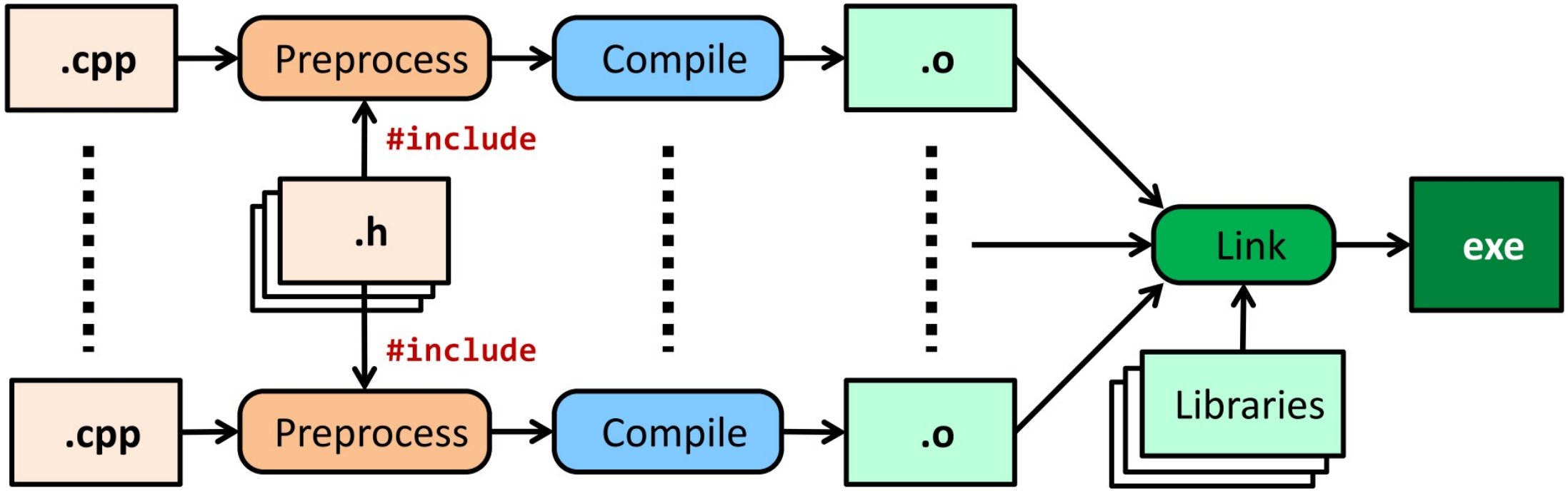
```
.method private specialname rtspecialname static
    void .cctor () cil managed
{
    // Method begins at RVA 0x2100
    // Code size 43 (0x2b)
    .maxstack 8

    IL_0000: ldstr "%d"
    IL_0005: newobj instance void class [FSharp.Core]Microsoft.FSharp.Core.PrintfFormat`5<class [FSharp
    IL_000a: stsfld class [FSharp.Core]Microsoft.FSharp.Core.PrintfFormat`4<class [FSharp.Core]Microsof
    IL_000f: call class [netstandard]System.IO.TextWriter [netstandard]System.Console::get_Out()
    IL_0014: call class [FSharp.Core]Microsoft.FSharp.Core.PrintfFormat`4<class [FSharp.Core]Microsoft.
    IL_0019: call !!0 [FSharp.Core]Microsoft.FSharp.Core.PrintfModule::PrintFormatLineToTextWriter<clas
    IL_001e: ldc.i4.5
    IL_001f: call valuetype [System.Private.CoreLib]System.Int32 _::sumOfSquares(valuetype [System.Priv
    IL_0024: callvirt instance !1 class [FSharp.Core]Microsoft.FSharp.Core.FSharpFunc`2<valuetype [Syst
    IL_0029: pop
    IL_002a: ret
```

CLR сама по себе ООПшная, ФП код на самом деле превращается в жесть

Компиляция и интерпретация на примере C++ и python

Процесс сборки на C++:

Препроцессинг (подключение директив) → Компиляция (преобразование в машинный код) → Линковка (исполняемый файл)



JIT-компиляция

- Язык компилируется не в машинный код, а в промежуточный.
- Во время выполнения JIT компилирует код в машинный.

Плюсы:

- Оптимизации недоступные при обычной компиляции
- Компиляция намного быстрее
- Не нужно заморачиваться с платформой, на которой будет исполняться код

Минусы:

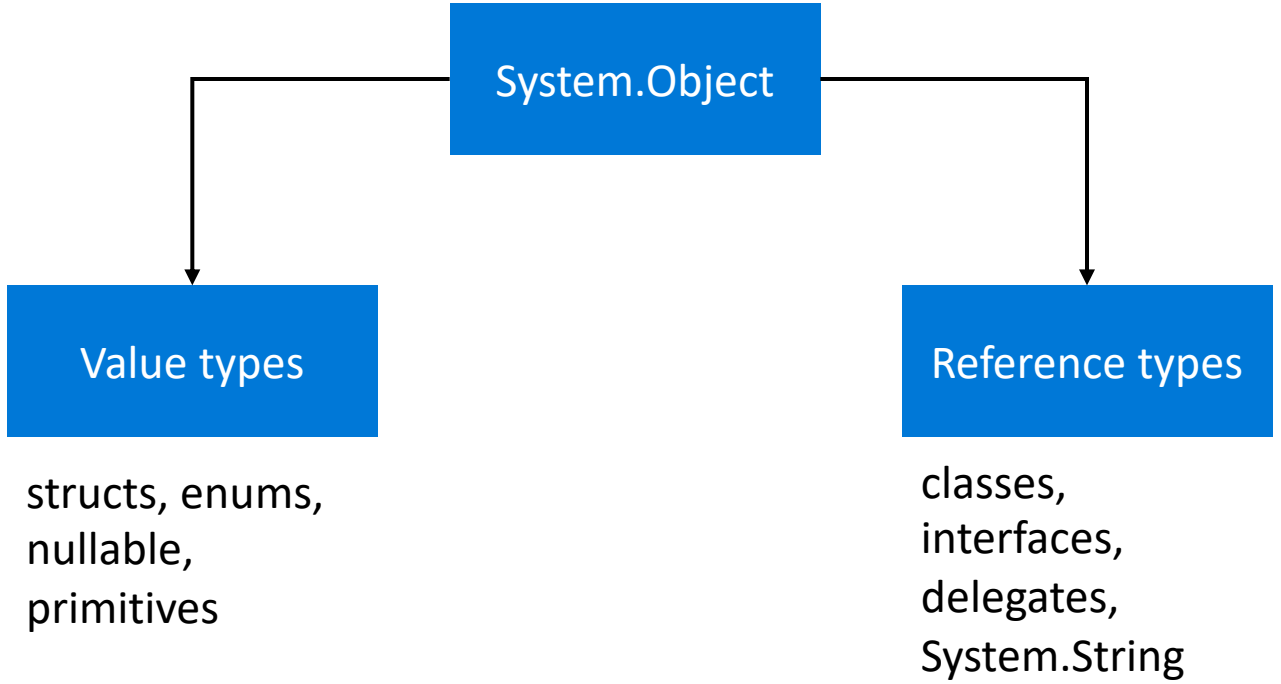
- Это не бесплатно - JIT компиляция имеет свой оверхед
- Медленный старт
- При дополнительных оптимизациях ОЧЕНЬ МЕДЛЕННЫЙ старт
- Декомпиляция

Garbage Collector (GC)

- Вам не нужно (но можно) руками очищать память как в C/C++
- Есть разные режимы для разных типов приложений
- В .NET не распространена практика кастомных GC как в JVM, но он не один
- Зато в .NET очень много способов не насиловать GC

Value Type, Reference Type, Boxing

- Value types, stack allocation
- Reference types, heap allocation
- Boxing – to heap, unboxing – to stack



Полезный тулинг

- github.com/dotnet (runtime, roslyn)
- source.dot.net
- sharplab.io
- dotMemory, dotTrace, dotPeek

Лабораторные работы (draft)

1. Hello world: Интероп, сравнение языков, пакетные менеджеры, бенчмарки, знакомство с dotnet runtime
2. Roslyn: написание анализатора кода и кодфиксера
3. Codegen: знакомство с инструментами кодогенерации
4. Perfomance: Гонки за миллисекунды
5. Zero allocation: Байтолюбство

Bonus: High performance GC friendly thread manager

Дополнительные баллы

- Контрибутинг в опенсорс позволяет получать дополнительные баллы.
- Количество баллов определяется в индивидуальном порядке.
- Контрибутить можно как в github.com/dotnet/runtime, так и в другие опенсорс-проекты (связанные с темой курса).
- Готовы помогать с контрибутами в dotnet/runtime, с другими проектами тоже, но лучше согласовывать свой выбор заранее.
- Контрибуты в документацию не считаются 😁