



Geçmişten Günümüze Java Mutfaktaki Projeler

Muhammed Fatih Doğmuş

- ODTÜ Bilgisayar Mühendisliği 2018
- TÜBİTAK BİLGEM YTE - Full-stack yazılım geliştirme mühendisi
-  fatihdogmus.dev
-  [@m_fatih_dogmus](https://twitter.com/m_fatih_dogmus)

Konuřmanın amacı

- Sürekli Java versiyonları çıkıyor
- Takip edebilmek için planlanan feature'ları bilmek gerekiyor

Projeler

**Projeler, büyük özelliklerin
geliştirildiđi yer**

Project lambda

- Brian Goetz önderliğinde
- Aralık 2009 – Şubat 2014
- Lambda functions, streams vs

Project Jigsaw

- Mark Reinhold önderliğinde
- Aralık 2008 – Eylül 2017
- Ana amacı JDK'i modüler hale getirmek
- Modül sistemi genel anlamda kullanılabilir

Biraz tebessüm 😊

- ‘Sun microsystems’in 2010’un ilk aylarında yayınlamayı planladığı JDK 7 versiyonundaki amacı, JDK’i modüler hale getirmektir.’ – Mark Reinhold

Linker

- [Project Jigsaw in JDK 9: Modularity Comes To Java](#)
- [Modules in One Lesson – Mark Reinhold](#)

Project Amber

Daha ufak, productivity iyileřtirmeleri

‘Eyvah, Kotlin geliyor’ 😄

| | Java 10 | Java 11 | Java 12 | Java 13 | Java 14 | Java 15 (*) |
|---|----------|----------|---------|-------------|-------------|-------------|
| Local-Variable Type Inference - var | Standard | | | | | |
| Local-Variable Syntax for Lambda Parameters | | Standard | | | | |
| Switch Expressions | | | Preview | 2nd Preview | Standard | |
| Text Blocks | | | | Preview | 2nd Preview | Standard |
| Records | | | | | Preview | 2nd Preview |
| Pattern Matching for instanceof | | | | | Preview | 2nd Preview |
| Sealed Classes | | | | | | Preview |
| More? | | | | | | |

Linkler

- [Java Language Futures - All Aboard Project Amber - Brian Goetz](#)
- [Evolving Java Project Amber - Vicente Romero](#)
- [Moving Java Forward Faster - Mark Reinhold](#)
- [Life Beyond Java 8 - Trisha Gee](#)
- [Java Versions and Features - Marco Behler](#)

Project Valhalla

Project Valhalla

- Brian Goetz liderliğinde ilerliyor
- Temmuz 2014'te başladı

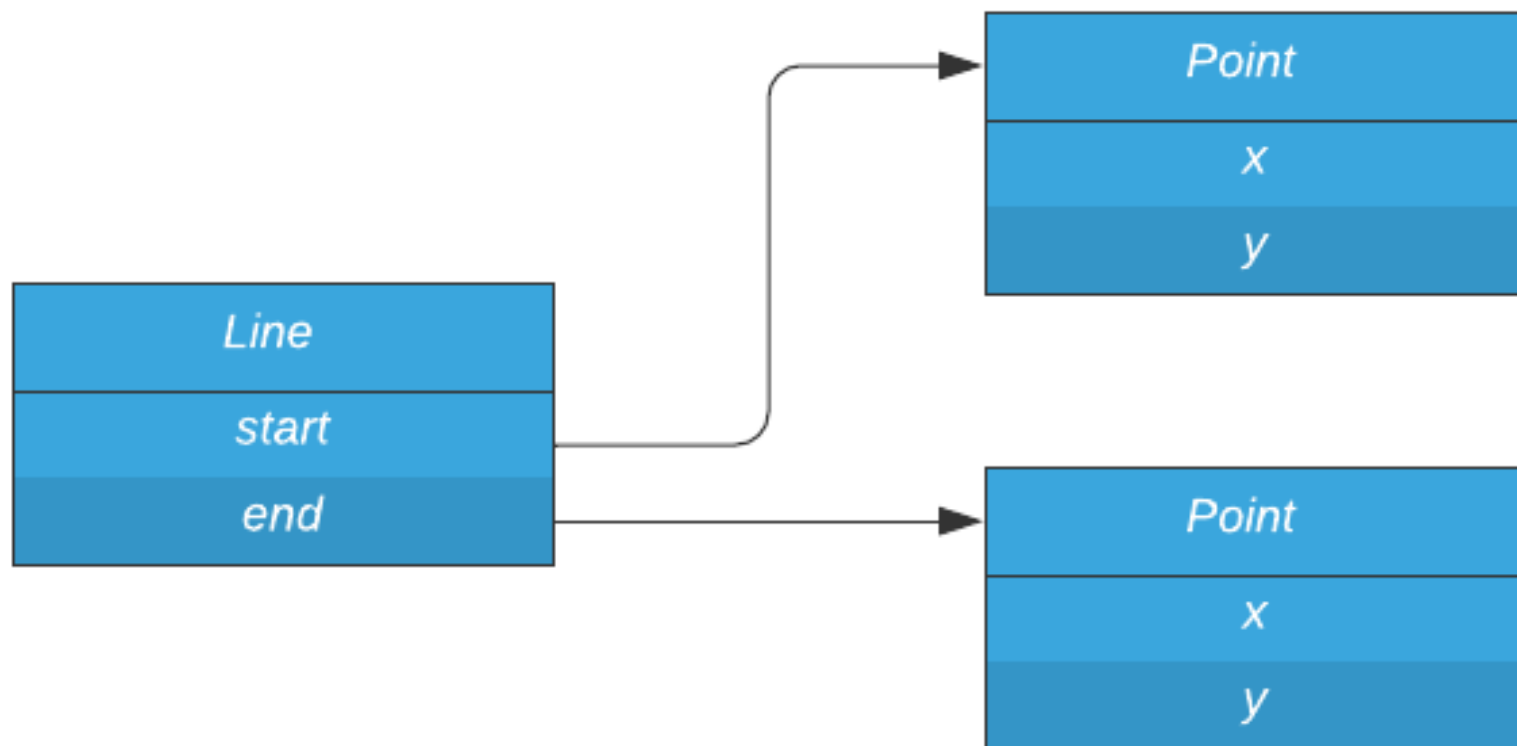
Java'da her şey bir object

- Mutable
- Memory access indirection
- Header için ekstra yer gerekiyor



```
public class Line {  
    private Point start;  
    private Point end;  
}
```

```
public class Point {  
    private double x;  
    private double y;  
}
```



Primitive types

‘Codes like a class, works like an int’

Primitive class

- Immutable
- Flat memory access
- Ekstra header yok



```
public primitive class Line {  
    private Point start;  
    private Point end;  
}
```

```
public primitive class Point {  
    private double x;  
    private double y;  
}
```

Line

start.x

start.y

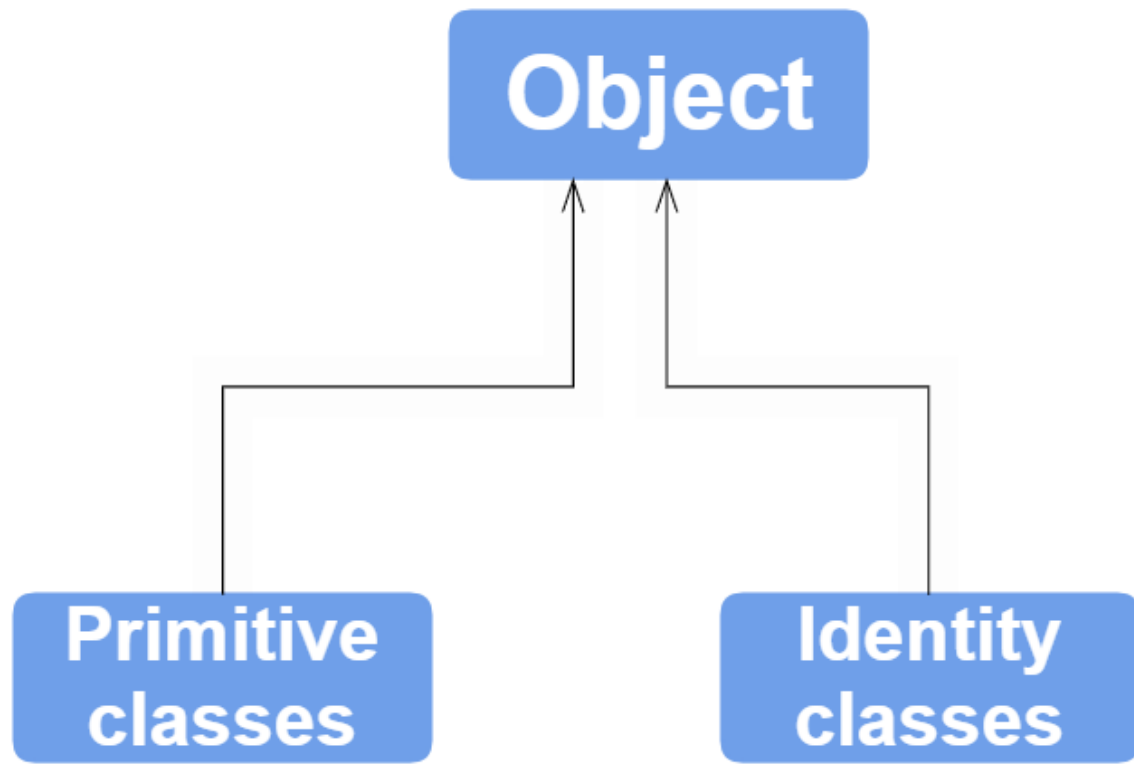
end.x

end.y

3 ayağı var

- Primitive classes
- Primitive'lerin obje hiyerarşisine getirilmesi
- Generic specilization

Primitive'ler, obje hiyerarşisine uymuyor



Primitives

Object

```
graph BT; PC[Primitive classes] --> O[Object]; IC[Identity classes] --> O; PC --- P[Primitives]; IC --- I[Identity classes]
```

**Primitive
classes**

Primitives

**Identity
classes**

Ne kazandırır?

- IntStream, DoubleStream gibi class'lar yok olur
- IntFunction, DoubleFunction gibi functional interface'lere ihtiyaç kalmaz
- İleride primitive'ler için ayrıyetten efor harcamaz

Specialized generics



```
List<int> intList = new ArrayList<>();
```

**Kendi primitive class'larımız için de
çalışır**

Linkler

- [Adventures on the Road to Valhalla – Brian Goetz](#)
- [Valhalla Update - Brian Goetz](#)
- [Project Valhalla: Bringing Performance to Java Developers](#)
- [PROJECT VALHALLA: FAST AND FURIOUS JAVA](#)
- [Value-Based Classes – Nicolai Parlog](#)

Project Loom

Project Loom

- Ron Pressler önderliğinde ilerliyor
- Ocak 2018'de başladı
- Java concurrency'yi baştan tanımlayacak

Java'da thread'ler

- Her bir Java thread'i, bir OS thread'i tarafından implement ediliyor
- Her thread çok fazla yer tutuyor
- Context switching ve thread oluřturması pahalı

Synchronous

- Her request için yeni bir thread

Avantajları

- Okuması ve yazması kolay
- Dil seviyesinde iyi desteği var(control flow, exceptions)
- Mevcut tool'lar il uyumlu(debugger, profiler)

Dezavantajları

- Çok masraflı

Synchronous

- Geliştirici 😊
- Donanım / OS 😞

**Kısmi çözüm olarak Thread Pool
kullanıyoruz**

Fakat bu da yeterli olmuyor

Asynchronous

- Bir istek, I/O için block'larsa, kullandığı thread, thread pool'a geri veriliyor
- I/O operasyonu bittiğinde, başka bir thread ile devam ediyor

Avantajları

- Çok daha scalable
- Çok daha iyi hardware utilization sağlıyor

Dezavantajları

- Okuması ve yazması çok zor

```
private Mono<?> fetchPersonsPublisher(Shell seaShell, SeaShellDto seaShellDtoReturn) {  
    return from(from(just(  
        seaShell.getPersons()).subscribeOn(parallel()).map(shellPersonRepository::findPersonsBlock).subscribeOn(parallel())  
        .map(persons -> {  
            persons.forEach(person -> seaShellDtoReturn.getPersons().add(SeaShellConverter.toShellPersonDto(person)));  
            return seaShellDtoReturn.getPersons();  
        })))  
        .thenMany(zip(  
            fetchPersonAccountPublisher(seaShellDtoReturn).subscribeOn(parallel()),  
            fetchPersonFullCostumePublisher(seaShellDtoReturn).subscribeOn(parallel()))));  
}
```

- Debug ve profil çıkarması(JFR, thread dump vs) çok zor
- Kurtulması imkansız

Asynchronous

- Geliştirici 🙄
- Donanım / OS 😄

‘Codes like sync, scales like async’

- Geliştirici 😊
- Donanım / OS 😊

Virtual thread

- User-space thread
- Context switch çok daha hızlı
- Çok daha az memory kullanıyorlar

>2_{KB} metadata

1_{MB} stack

200-300_B metadata

Pay-as-you-go stack

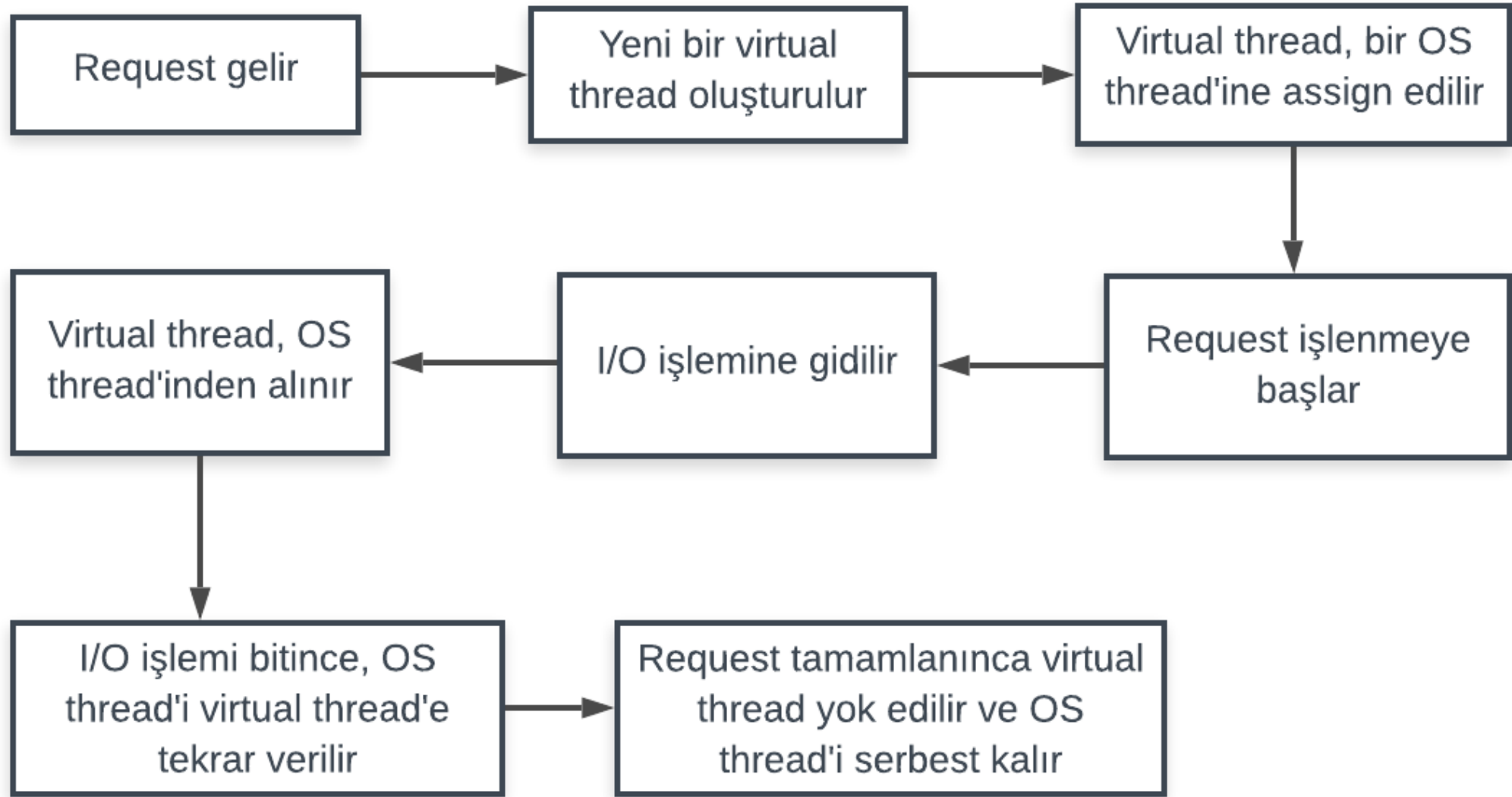
1-10_{μs}

~200_{ns}

virtual threads



'carrier' thread'ler



```
Thread t = Thread.startVirtualThread(() -> {  
    System.out.println("Hello, Loom!");  
});
```

```
Thread t = Thread.builder().virtual().task(() -> { ... }).build();
```

```
Thread t = Thread.builder().virtual().task(() -> { ... }).start();
```

```
ThreadFactory factory = Thread.builder().virtual().factory();
```

- [What is Project Loom? An introduction – Adam Warski](#)
- [Project Loom: Modern Scalable Concurrency for the Java – Ron Pressler](#)
- [Inside Java Episode 8 - “Project Loom” with Ron Pressler](#)
- [Will Project Loom obliterate Java Futures?](#)
- [Do Loom’s Claims Stack Up? Part 1: Millions of Threads?](#)
- [Do Looms Claims Stack Up? Part 2: Thread Pools?](#)

Project Panama

Tüm yazılan kütüphaneler, Java ile yazılmamıştır

Native kütüphaneler

- Deep learning kütüphaneleri: TensorFlow, Theano, Torch
- GPU programlaram kütüphaneleri: CUDA, OpenCL
- Grafik kütüphaneleri: OpenGL, Vulkan, DirectX

Bugünün çözümü: JNI

Gereksiz yere native kod yazmamız lazım

Project panama, bunu çözmeye geliyor

3 katmanlı bir çözüm sunuyor

Foreign Memory API

Foreign Linker API

```
import java.lang.invoke.*;
import jdk.incubator.foreign.*;

class PanamaMain {
    public static void main(String[] args) throws Throwable {
        // get System linker
        var linker = CLinker.getInstance();
        var lookup = LibraryLookup.ofDefault();

        // get a native method handle for 'getpid' function
        var getpid = linker.downcallHandle(
            lookup.lookup("getpid").get(),
            MethodType.methodType(int.class),
            FunctionDescriptor.of(CLinker.C_INT));

        // invoke it!
        System.out.println((int) getpid.invokeExact());
    }
}
```

jextract


```
$ jextract -t com.unix getpid.h
```

```
WARNING: Using incubator modules: jdk.incubator.foreign, jdk.incubator.jextract
```

```
import static com.unix.getpid_h.*;
```

```
class Main2 {
```

```
    public static void main(String[] args) {
```

```
        System.out.println(getpid());
```

```
    }
```

```
}
```

Linker

- [Project Panama and jextract](#)
- [Project Panama's Foreign API](#)
- [Panama Update with Maurizio Cimadamore](#)
- [Inside Java Podcast 9 - "Project Panama - The Foreign Memory Access API"](#)
- [Inside Java Podcast 10 - "Project Panama - The Foreign Linker API"](#)

Dinlediđiniz iin teŕekkür ederim

İletişim adresleri

 fdogmus06@gmail.com

 [@m_fatih_dogmus](https://twitter.com/m_fatih_dogmus)

 [Muhammed Fatih Doğmuş](https://www.linkedin.com/in/Muhammed-Fatih-Doğmuş)

 fatihdogmus.dev