



# کارگاه برنامه نویسی پیشرفته

آرش مغانی طرقي  
[moghani.arash@gmail.com](mailto:moghani.arash@gmail.com)



# داتین

- راهکارهای جامع بانکی
- ۱۵۰۰ همکار در سراسر ایران
- دفترهای تهران، مشهد، اصفهان، شیراز (بزودی)
- ۲۰۰ همکار در مشهد
- مدرسه‌ی داتین
- همکاری با دانشگاهها
- بانکهای پاسارگاد و سپه و رسالت و شهر



# هدف این درس

- آشنایی با ابزارهای موجود برای بهبود کدهای خروجی
- آشنایی با فرآیندهای تولید یک نرم افزار
- انجام یک پروژه‌ی عملیاتی از ابتدا تا انتها برای درک فضای موجود در محیط کاری و شرکتها



# آنچه می آموزیم

- سیستم‌های مدیریت سورس کد (VCS)
- کد نویسی جاوا با تمرکز بر برخی مفاهیم مانند IoC و DI
- توسعه‌ی آزمون محور
- فریمورک اسپرینگ و اسپرینگ بوت
- ساختار شبکه و مفهوم REST
- داکر و داکر کامپوز
- پارادایم AOP



# پیش نیازها

- آشنایی با متدولوژی‌های تولید نرم افزار
- تسلط بر یک زبان برنامه نویسی سمت سرور مانند جاوا یا پایتون
- آشنایی با سیستم عامل و خط فرمان

# پروژه‌ی پایان‌ترم

- هدف از پروژه، پیاده‌سازی مفاهیم گفته شده در طول ترم است.
- پروژه به انتخاب دانشجو بوده و هدف، ایجاد یک سرور برای سرویس‌دهی به وسیله‌ی Restful API به زبان جاوا و با استفاده از اسپرینگ بوت است. (در صورتی که پروژه‌ی انتخابی شما دارای سرویس‌دهی به وسیله‌ی اندپوینت‌های رستفول نیست، نیز باید برای بخشی از سیستم برای مثال بخش مدیریت سیستم از Restful API استفاده شود).
- استفاده از ORM، داکر، یونیت تست نویسی و همچنین نوشتن داکيومنت برای اندپوینت‌های نوشته شده توسط ابزارهای موجود مانند سواگر مورد نظر است.
- ایجاد داکيومنت برای خود کد به وسیله‌ی JavaDocs یا هرگونه ابزار اتوماتیک نیز دارای امتیاز مثبت می‌باشد.



# پروژه‌ی پایان‌ترم

■ صرف استفاده از مفاهیم گفته شده مورد نظر می‌باشد و هدف حجیم شدن پروژه نیست و استفاده از مفاهیم گفته شده به صورت افراطی نیست. برای مثال در صورت نوشتن جاوا داکز هدف نوشتن داکيومنت برای کل پروژه نیست و نوشتن داکيومنت برای بخشی از پروژه برای یادگیری مفاهیم گفته شده کفایت می‌کند.

# ایده های پروژه

- لیست کارها: یک اپلیکیشن مدیریت کارها با عملیات CRUD.
- اپلیکیشن حسابداری: هزینه های روزانه را ردیابی می کند و خلاصه ای از آنها می دهد.
- یافتن دستور غذا: با استفاده از API، دستورات غذایی مرتبط با مواد موجود را پیدا کند.
- جمع آوری خبر: اخبار را از منابع مختلف جمع آوری کند.
- تبدیل ارز: با استفاده از API، ارزها را به صورت زنده تبدیل کند.
- کوتاه کننده لینک: URL های طولانی را کوتاه کرده تا به راحتی به اشتراک گذاشته شوند.



# ایده های پروژه (ادامه)

- اپلیکیشن فلش کارت: برای یادگیری زبان ها یا موضوعات جدید.
- سازنده میم: به کاربران اجازه می دهد تا متن را به تصاویر میم محبوب اضافه کنند.
- ردیاب عادت ها: یک برنامه وب که به کاربران اجازه می دهد تا عادت های روزانه خود را پایش و مدیریت کنند.
- ربات تلگرام:
  1. دانلود از یوتیوب و اینستاگرام
  2. آپلود فایل (لینک داده شده در تلگرام آپلود می شود)
  3. ربات chatgpt (با استفاده از API و دارای محاسبه ی قیمت)

# سیستم نمره دهی پروژه

■ تمرینات کلاسی

■ پایان ترم: تحویل پروژه در انتهای ترم

1. نمره ی اضافه:

هرگونه خلاقیت در انجام پروژه ها

مستندسازی جامع کد با استفاده از ابزارهای موجود مانند JavaDocs و

Swagger

تست نویسی برای حالت های خاص

ایجاد داکر فایل کم حجم

# ساختار یک تیم نرم افزاری





# جاوا چیست؟ چرا جاوا؟

■ بالغ بر ۱۳۰۰ زبان برنامه نویسی در جهان داریم

■ <https://www.tiobe.com/tiobe-index/>

■ طبق ویکی پدیا ۷۱۶

■ [https://en.wikipedia.org/wiki/List\\_of\\_programming\\_languages](https://en.wikipedia.org/wiki/List_of_programming_languages)



# سان مایکروسیستم

■ سرورهای سخت افزاری

■ سولاریس OS

■ جاوا

■ [https://en.wikipedia.org/wiki/Sun\\_Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems)

■ NetBeans IDE

■ MySQL

■ Glassfish



# بنیاد نرم افزاری جاوا

- Java Community Process (<https://jcp.org>)
- Apache Software Foundation (ASF) ([apache.org](https://apache.org))
- Eclipse Foundation
- Google
- Oracle
- IBM
- Red Hat
- Google
- JetBrains



# Java Specification Request (JSR)

- <https://jcp.org/en/jsr/all>
- JSR 14: Add Generic Types To The Java Programming
- JSR 54: Java Database Connectivity 2.0 API Specification
- JSR 121: Application Isolation API Specification
- JSR 133: Java Memory Model and Thread Specification
- JSR 160: Java Management Extensions (JMX) Remote API
- JSR 250: Common Annotations for the Java Platform:

# Java duke





# نسخه‌های جاوا

- [https://en.wikipedia.org/wiki/Java\\_version\\_history](https://en.wikipedia.org/wiki/Java_version_history)
- **Java 5 (2004):** Introduced Generics, annotations, autoboxing/unboxing, enhanced for loop, varargs, static imports, concurrency utilities.
- **Java 6 (2006):** Scripting support, JDBC 4.0, Java Compiler API, improvements in JVM performance and Java GUI.
- **Java 7 (2011):** Try-with-resources, diamond operator, NIO.2 file API, Fork/Join Framework, binary literals, switch with strings.
- **Java 8 (2014, LTS):** Lambda expressions, Stream API, new Date-Time API, default methods in interfaces, Nashorn JavaScript engine.
- **Java 11 (2018, LTS):** Standard HTTP Client, new Epsilon garbage collector, Flight Recorder, Nest-Based Access Control.
- **Java 17 (2021, LTS):** Records (standard), Pattern Matching for instance of (standard), Sealed classes (standard).



# جاوای خالص

- جاوا وانیلا یا جاوای خالص
- چارچوب کاری اسپرینگ
- ایجاد یک برنامه جاوا (تک فایل) بدون IDE  
1. اجرای فایل کلاس بدون زیپ کردن (Exploded)
- ایجاد یک فایل jar کامل بدون IDE
- ایجاد یک پروژه با استفاده از Maven

# انواع داده در جاوا

## Primitive Types:

1. **byte:** 8-bit signed integer.
2. **short:** 16-bit signed integer.
3. **int:** 32-bit signed integer.
4. **long:** 64-bit signed integer.
5. **float:** 32-bit floating point.
6. **double:** 64-bit floating point.
7. **char:** 16-bit Unicode character.
8. **boolean:** Represents true/false values.

## Reference Types:

1. **Class types:** Represent objects of a class
2. **Interface types:** Represent interfaces.
3. **Array types:** Represent arrays of a specified type, like `int[]`, `double[]`, `Object[]`, etc.



# ایجاد اولین پروژه در جاوا

- یک پروژه‌ی Hello world ایجاد کنید.
- تاریخ و زمان فعلی سیستم را در صفحه چاپ کنید.
- یک تابع برای بررسی Palindrome بودن رشته بنویسید.
- رشته‌ی پالیندروم: به رشته‌ای از حروف گفته می‌شود که از دو طرف (راست به چپ و چپ به راست) دقیقاً به یک شکل خوانده می‌شوند. (در واقع از دو طرف متقارن هستند).
- تمرین: ورودی تابع پالیندروم را از ورودی سیستم دریافت کنید.



# توسعه‌ی آزمون محور

■ Test Driven Design (TDD)

■ توسعه آزمون محور: نوشتن تست‌ها قبل از نوشتن کد اصلی.

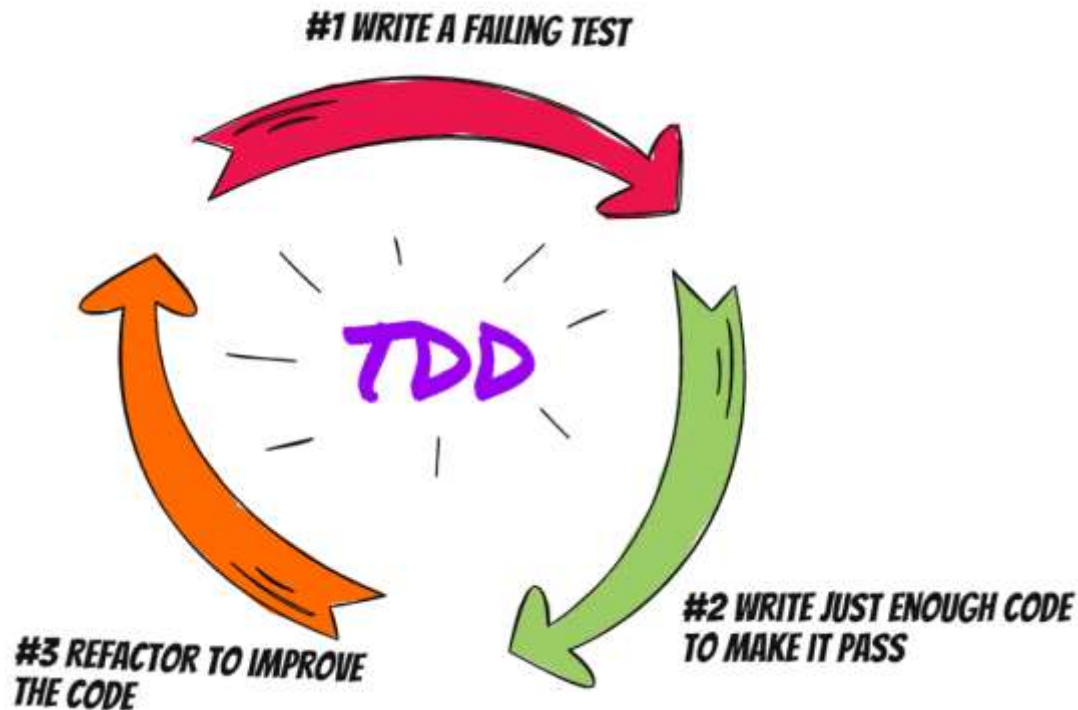
■ چرخه قرمز-سبز-بازنویسی:

1. قرمز: نوشتن یک تست ناموفق.

2. سبز: نوشتن حداقل کد لازم برای قبولی تست.

3. بازنویسی: بهبود کد ضمن حفظ قبولی تست‌ها.

# توسعه‌ی آزمون‌محور





# اضافه کردن دیندنسی

- استفاده از Maven و Gradle
- جستجو در سایت [mvnrepository.com](https://mvnrepository.com)
- استفاده از یونیت تست در نوشتن کدها با استفاده از Junit
- **تمرین کلاسی:** تستهای مورد نیاز برای تست کردن تابع پالیندروم نوشته شده را بنویسید. (با نادیده گرفتن فاصله‌ها، نشانه‌گذاری‌ها و حروف بزرگ یا کوچک)



# چست؟ REST

- REpresentational State Transfer
- Statelessness
- User Http verbs
- HTTP status codes
- Cacheability





# تست API

- **Postman:** ابزار پستمن برنامه‌ای است که برای تست API استفاده می‌شود. پستمن یک درخواست http را به وب سرور ارسال می‌کند و پاسخ را دریافت می‌کند.
- **Curl:** کرل یک ابزار خط فرمان است که برای ارسال و دریافت داده‌ها با استفاده از پروتکل‌های مختلف اینترنتی مانند HTTP, HTTPS, FTP و غیره استفاده می‌شود. این ابزار به شما امکان می‌دهد که درخواست‌های وب را از خط فرمان ارسال کنید و پاسخ‌های سرور را دریافت کنید.

# نسخه‌های جاوا

- **جاوا SE (Standard Edition):** این نسخه استاندارد جاوا بوده و شامل کتابخانه‌ها و API‌های اصلی برای ساخت برنامه‌های دسکتاپ، سرور و اپلیکیشن‌های کوچکتر می‌باشد.
- **جاوا EE (Enterprise Edition):** این نسخه از جاوا برای توسعه برنامه‌های بزرگ مقیاس سازمانی و وبسایت‌های پیچیده طراحی شده است. جاوا EE شامل مجموعه‌ای از API‌ها و کتابخانه‌های خاص برای توسعه وب‌اپلیکیشن‌ها، سرویس‌های وب، مدیریت دیتابیس و بسیاری امکانات دیگر است.
- **جاوا SE** بیشتر برای توسعه برنامه‌های عمومی و کوچکتر استفاده می‌شود.
- **جاوا EE** برای پروژه‌های سازمانی بزرگ و وبسایت‌هایی با نیازهای فنی پیچیده مناسب است.

# اسپرینگ بوت

- **اسپرینگ فریمورک (Spring Framework):** یک چارچوب نرم‌افزاری و مدیر وارونگی کنترل متن‌باز برای سکوی جاوا است. ویژگی‌های اصلی این فریمورک می‌تواند توسط هر برنامه جاوا مورد استفاده قرار گیرد، اما دارای افزونه‌هایی برای ساختن برنامه‌های کاربردی وب بر روی پلت فرم JavaEE می‌باشد.

- **اسپرینگ بوت:** پیکربندی اسپرینگ

1. ایجاد پروژه‌ی اسپرینگ بوت: `start.spring.io`
2. از طریق IntelliJ IDEA



# ورژن جاوا و اسپرینگ

- **3.2.1:** This is the latest milestone version of Spring Boot, released on July 20, 2023. It requires Java 17 as a minimum version and is compatible with Spring Framework 6.0.12 or above.
- **2.7.17:** This is the latest maintenance release of Spring Boot 2.7.x, released on November 18, 2021. It requires Java 8 or 11 as a minimum version and is compatible with Spring Framework 5.3.18 or above.

# داکر چیست؟

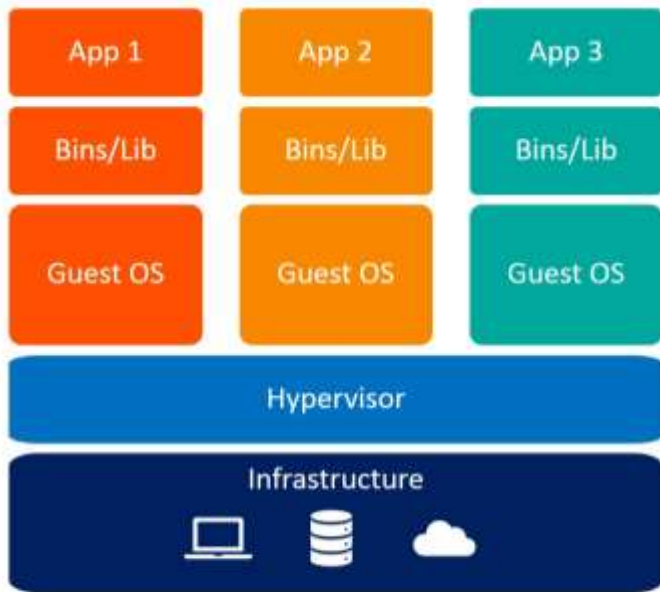
■ داکر یک پلتفرم متن‌باز است که برای توسعه، تحویل و اجرای برنامه‌ها با استفاده از مفهوم کانتینرها طراحی شده است. کانتینرها به توسعه‌دهندگان این امکان را می‌دهند که برنامه‌ها را در محیطی ایزوله بسته‌بندی کنند به طوری که برنامه به راحتی و به طور یکسان در هر محیطی قابل اجرا باشد.



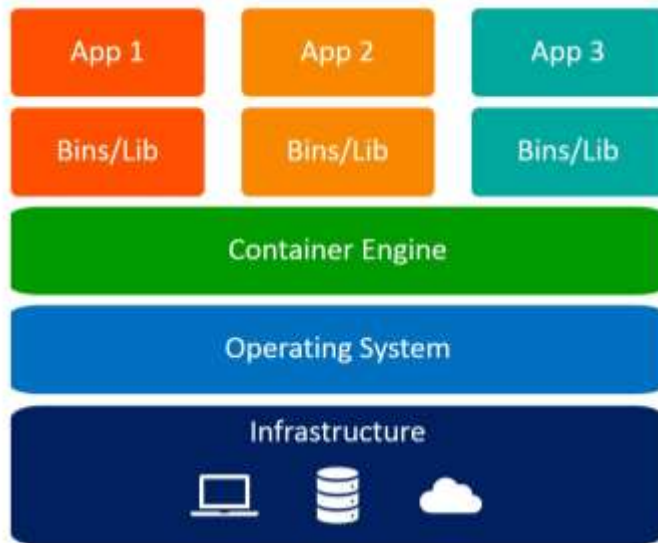
# داگر چیست؟

- ماشین‌های مجازی: دارای سیستم‌عامل کامل مستقل خود هستند و معمولاً کندتر عمل می‌کنند.
- کانتینرها: از سیستم‌عامل میزبان به اشتراک گذاشته شده استفاده می‌کنند و معمولاً سریع‌تر هستند.

# ساختار داکر

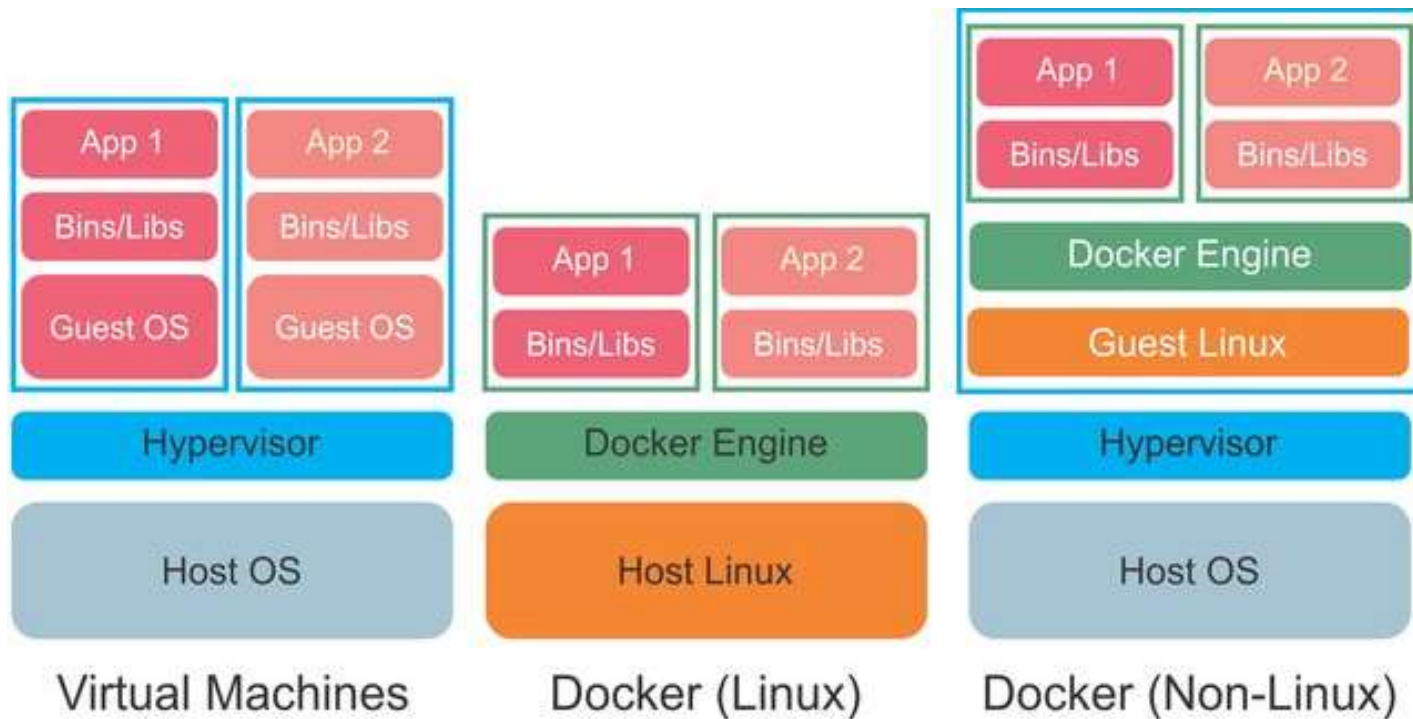


Virtual Machines



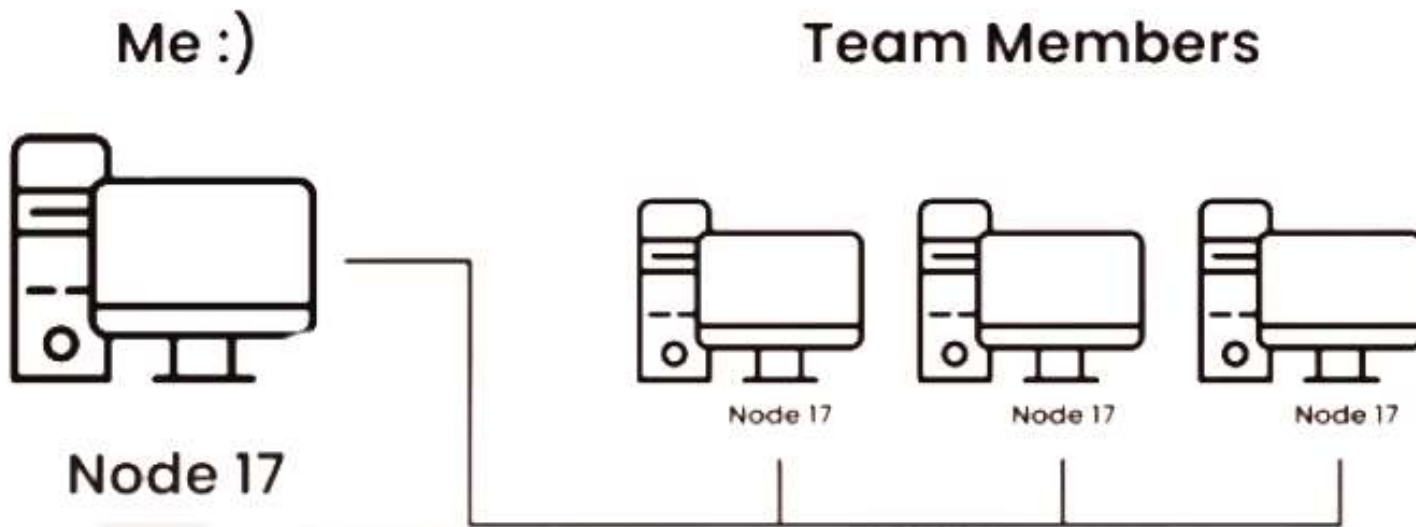
Containers

# ساختار داکر





# مزایای داکر - همکاری تیمی





# مزایای داکر – چند ابزار روی یک سیستم

## Containers

Node 17  
dependencies  
source code

Node 15  
dependencies  
source code

Python 3  
dependencies  
source code

# نصب داکر

- Linux:
  - <https://docs.docker.com/engine/install/ubuntu/#install-using-the-convenience-script>
- Windows:
  - <https://docs.docker.com/desktop/install/windows-install/>
  - Current latest version as of 2023/12/26 is Docker Desktop 4.26.1
  - Task manager > performance tab > virtualization > enabled

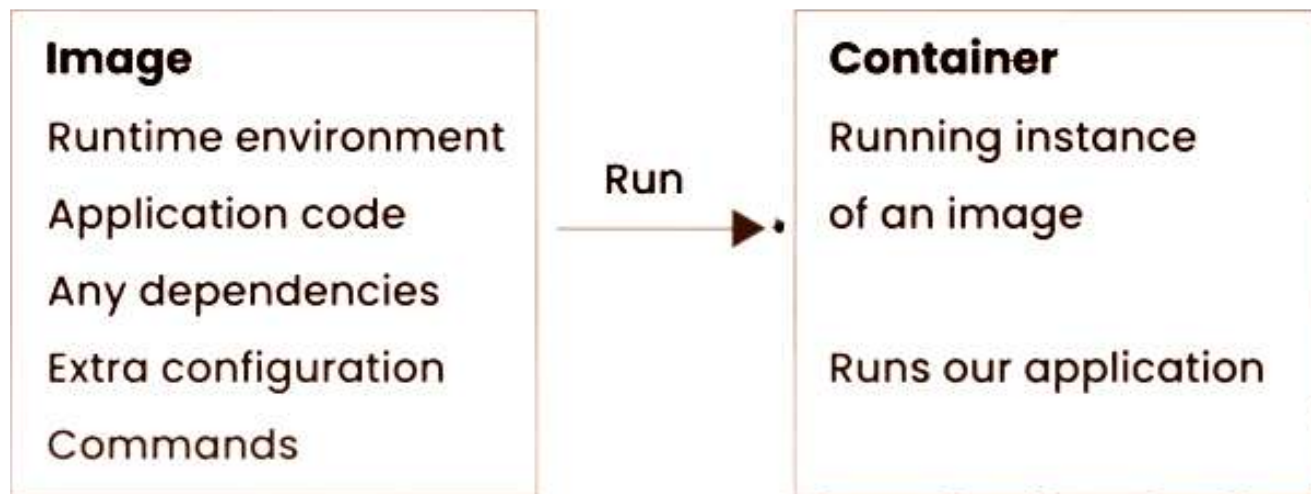


# داکر ایمیج

- نمونه‌ی اولیه‌ای برای اجرای کانتینر
- محیط اجرای کد
- کد اپلیکیشن
- دپندنسی‌های کد
- تنظیمات (شامل متغیرهای env سیستم عامل)
- فرمانها

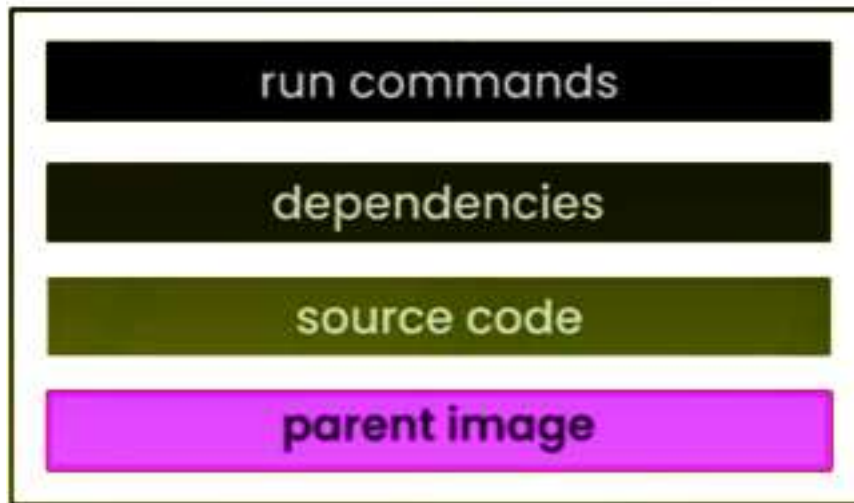
# کانتینر

اجرا شده‌ی یک داکر ایمیج



# لایه بندی داکر ایمیج

## Image



# فرمانهای پایه

لیست ایمیجها

`docker images`

گرفتن یک ایمیج

`docker pull hello-world`

اجرای یک ایمیج و ایجاد کانتینر

`docker run hello-world`

لیست کانتینرها

`docker ps -a`

# ایجاد داکر فایل

- create Dockerfile
- `docker build -t college .`
- `docker run -d college`
- `docker ps -a`
- `docker run --name college_c1 -p 8082:8082 -d --rm college`



# داکر کامپوز

- `docker-compose.yml`
- `Docker-compose up -d`
- `Docker-compose down -d`



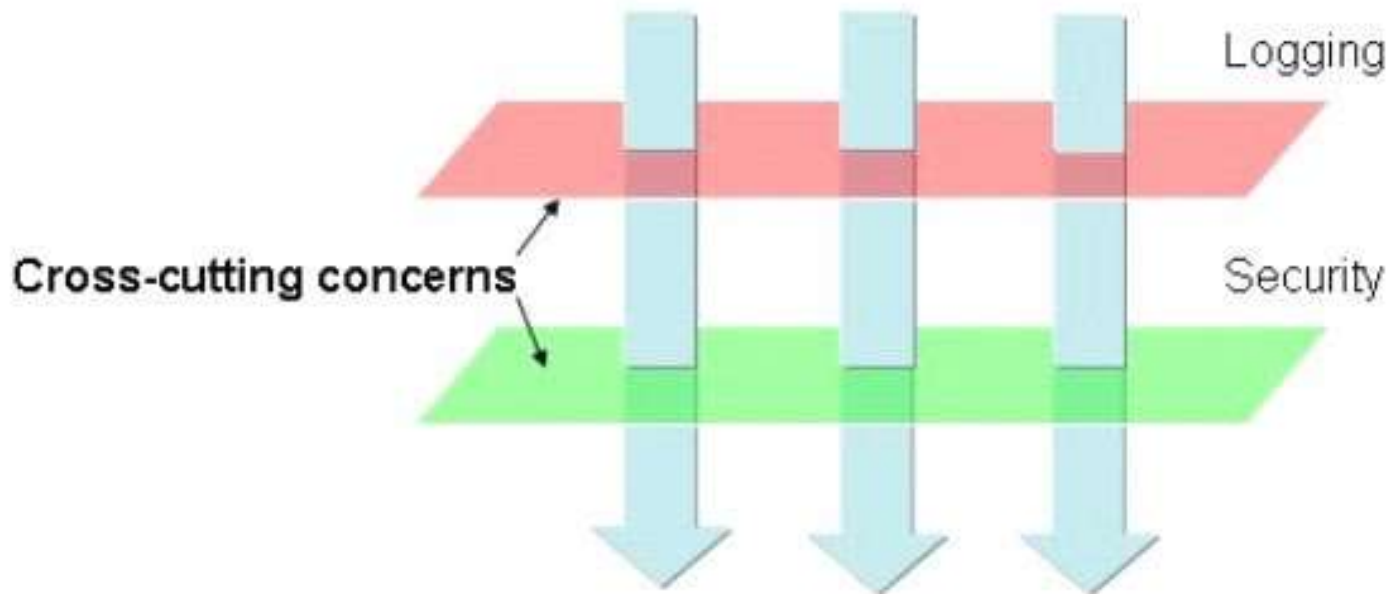
# بهینه سازی داکر فایل

1. استفاده از تصاویر پایه رسمی و حداقلی
2. کش کردن لایه‌ها
3. کاهش تعداد لایه‌ها
4. حذف فایل‌های غیرضروری
5. استفاده از `dockerignore`.

# پارادایم‌های برنامه نویسی

Paradigm	JavaScript	Java	Python	C	C++
Object-Oriented	Prototype-based	Strongly supported	Supported	Not supported	Strongly supported
Functional	Supported	Supported (Java 8+)	Supported	Limited support	Supported (C++11+)
Procedural	Supported	Supported	Supported	Strongly supported	Supported
Aspect-Oriented (AOP)	Via libraries	Strongly supported (Spring AOP, AspectJ)	Via libraries (AspectLib)	Not common	Via libraries

# دغدغه‌های متقاطع



# دغدغه‌های متقاطع

@Service

public class Business1 {

3 usages

private final Logger logger = LoggerFactory.getLogger(this.getClass());

1 usage    Arash\_M \*

public void calculateSomething() throws InterruptedException {

logger.trace("Starting execution of {}", getClass().getName()); //cross-cutting logging

//Main calculation

Thread.sleep( millis: 500);

logger.info("Hello world");

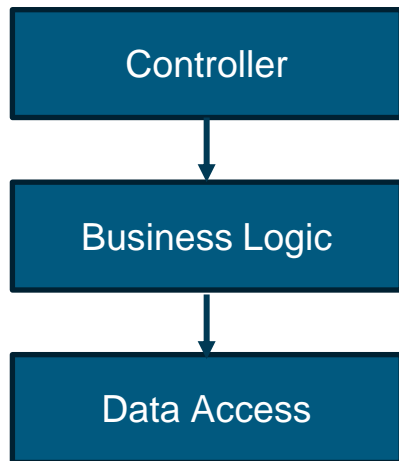
logger.trace("End of execution of {}", getClass().getName()); //cross-cutting logging

}

}

# دغدغه‌های متقاطع

Layers:



- امنیت
- احراز هویت
- لاگ
- مدیریت تراکنش دیتابیس

# مزیتها

- Cross Cutting Concerns
- Reuse
- Quick Development
- Focus on one aspect
- Enabled/Disabled

# برنامه نویسی جنبه گرا (AOP)

- برنامه نویسی جنبه گرا: AOP یا همان Aspect-Oriented Programming یک پارادایم برنامه نویسی است که به توسعه دهندگان اجازه می دهد دغدغه های متقاطع مانند نمایش لاگ و بخش امنیت را در برنامه های خود مازول بندی کنند.
  - Spring AOP: یک چارچوب قدرتمند برای پیاده سازی AOP در برنامه های جاوا است که به خوبی با چارچوب محبوب Spring یکپارچه شده است.
  - AOP در جاوای خالص: برای این کار از AspectJ استفاده می شود.
- AspectJ یک ابزار قدرتمند برای اجرای برنامه نویسی جنبه گرا در سطح کامپایلر است که امکان افزودن جنبه ها به کد جاوا را قبل از اجرای آن فراهم می کند. برای استفاده از AspectJ در پروژه های مبتنی بر Maven، باید دیپندنسی هایی مانند aspectjrt و aspectjweaver به فایل pom.xml اضافه شوند.



# مفاهیم AOP

- پوینت کات (Pointcut): پوینت کات عبارتی است که برای تعریف زمانی که یک فراخوانی به متد باید قطع شود استفاده می شود.
- آدوایس (Advice): کاری است که قرار است انجام شود. منطقی که می خواهید هنگام قبل یا پیش از یک متد اجرا کنید.
- جنبه (Aspect): ترکیبی از تعریف اینکه چه زمانی می خواهید یک فراخوانی متد را قطع کنید (پوینت کات) و چه کاری انجام دهید (آدوایس)، اسپکت نامیده می شود.
- نقطه ی اتصال (Join Point): وقتی کد اجرا می شود و شرایط پوینت کات برآورده می شود، آدوایس اجرا می شود. جوینت پوینت نقطه ی اجرای یک آدوایس است.
- ویور (Weaver): ویور چارچوبی است که AOP را پیاده سازی می کند. مانند: AspectJ یا Spring AOP



پرسش و پاسخ



تشکر  
از توجه شما