



# تحلیل و طراحی سیستم‌ها

## بخش 6 طراحی معماری نرم‌افزار

(برگدان اسلامیدهای (توسط دانشجویان دانشگاه علم و فرهنگ)

- J. S. Valacich, J. George, Modern Systems Analysis and Design. 8th Edition, Pearson 2017.
- I. Sommerville. Software Engineering. 10th Edition, Pearson, 2016.

# مباحث تحت پژوهش

- تصمیمات معماری طراحی
- دیدگاه‌های معماری
- الگوهای معماری
- معماری برنامه

# طراحی معماری

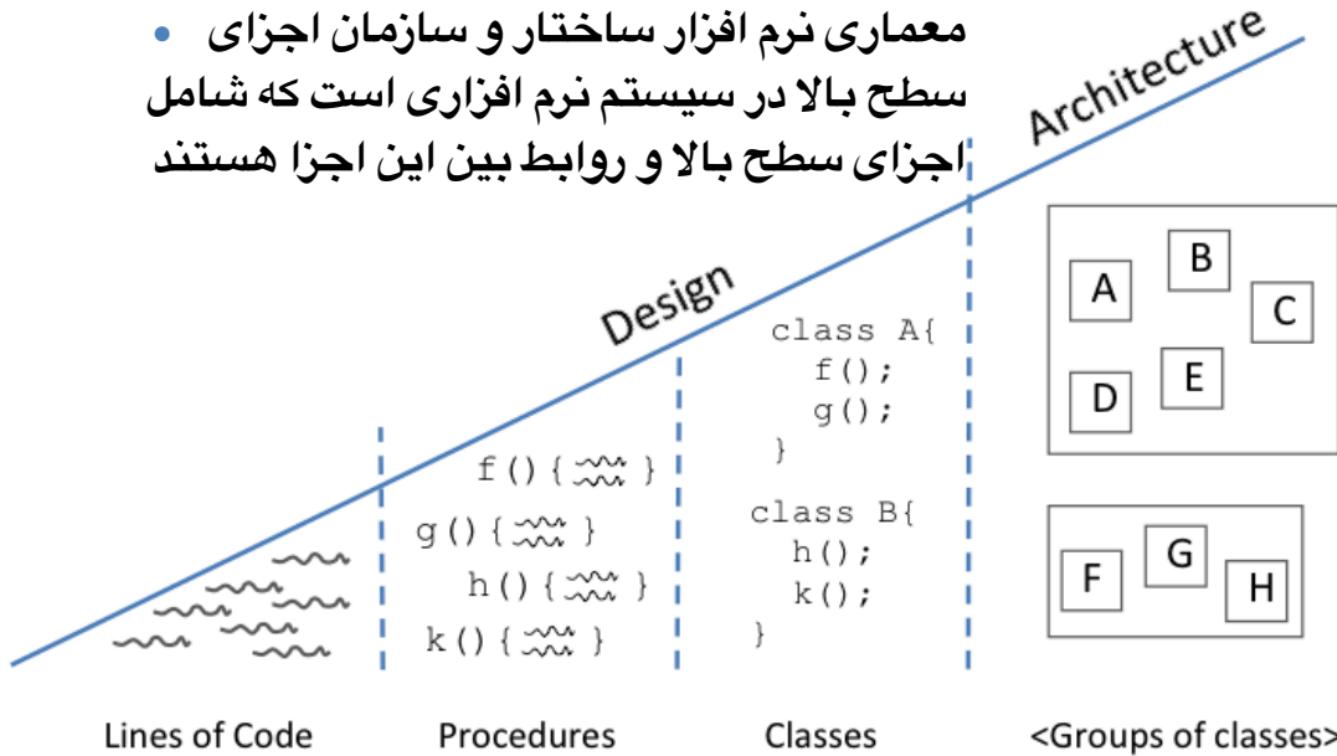
- طراحی معماری نگرانی در مورد چگونگی سازماندهی سیستم.  
نرم افزاری و طراحی ساختار کلی این سیستم است.  
طراحی معماری ارتباط حیاتی بین طراحی و مهندسی الزامات  
است؛ زیرا اجزای اصلی ساختار در یک سیستم و روابط بین آنها  
(زیرسیستم ها، برنامه ها یا کلاس های کلیدی) را مشخص می  
کند.
- خروجی فرآیند طراحی معماری یک مدل معماری است که توصیف  
نحوه سازماندهی سیستم را به عنوان مجموعه ای از اجزای  
ارتباط برقرار می کند.
- به طور کلی پذیرفته شده است که یک مرحله اولیه از فرآیندهای  
چابک، طراحی یک سیستم کلی سیستم می باشد

# طبقه بندی معماری

- معماری سیستم
- معماری نرم افزار
- شرکت معماری
- معماری کسب و کار
- ساختار اطلاعات
- معماری برنامه
- معماری داده
- فناوری / معماری زیرساخت
  - معماری مرجع
  - معماری خط تولید

# نقش معماری

- معماری نرم افزار ساختار و سازمان اجزای سطح بالا در سیستم نرم افزاری است که شامل اجزای سطح بالا و روابط بین این اجزا هستند



Lines of Code

Procedures

Classes

<Groups of classes>

# مثال

- خانه‌ی وینچستر راز در سن خوزه کالیفرنیا طراحی عالی ((طرح ایجاد)، اما معماری ضعیف (طرح ساختار

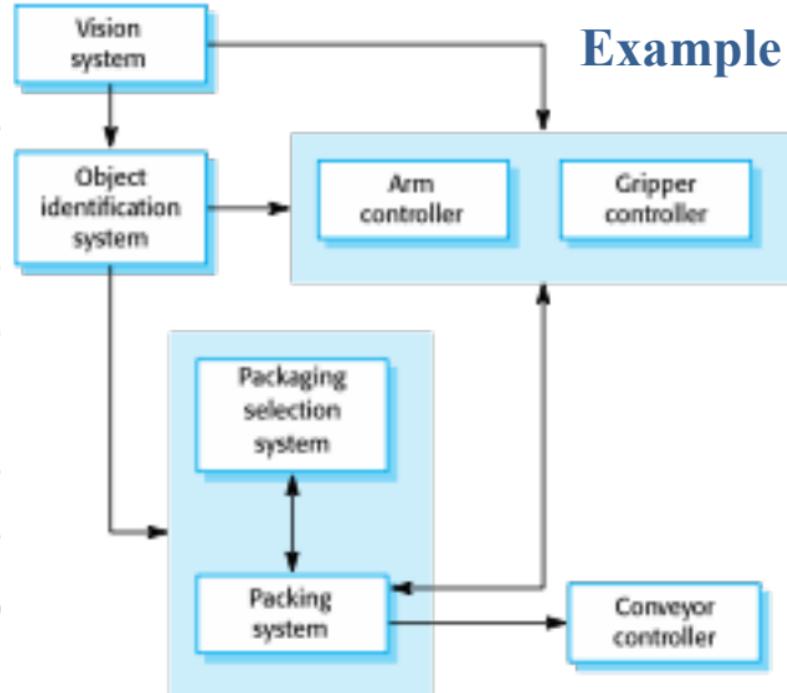


# اسرار خانه وینچستر

- سارا وینچستر، بیوه ای از سرباز اسلحه ویلیام ویرت وینچستر، پس از مرگ شوهرش مقدار زیادی پول را به ارث برده است او مقاعد شد که خانواده اش توسط ارواح همه کسانی که توسط خانواده های ساخته شده اسلحه کشته شده بودند، نفرین شده بودند و تنها راهی که برای جلوگیری از این ارواح شیطانی بود، این بود که به طور مداوم بر روی خانه اش ساخته شود.
- این همان چیزی است که او طی 38 سال از سال 1884 تا زمان مرگش در سال 1922 با بیش از 20.5 میلیون دلار (معادل 520 میلیون دلار در سال 2017) انجام داده است.
- مشکل او این نبود که یک معمار را برای نظارت بر تغییرات استخدام کند. تغییراتی که او به خانه انجام داده بود همه محلی بودند.
- نتیجه: پله هایی که به جایی نمی رستند، مکان های آتش نشانی بدون شدن هایی هستند که بخش ۶ مادجه، معنای آنها افزایش شده باشد.

## Example

- مشتری / WWW معماری سرور است.
  - دو نوع اجزای تشکیل دهنده سبک معماری WWW: عبارتند از WWW مشتریان = مرورگرهایی مانند فایرفاکس
  - سرور = سرورهای وب مانند Apache



The architecture of a packing robot control system

# انتزاع معماری

معماری کوچک، با معماری برنامه های فردی مرتبط است. در این سطح، ما به نحوی است که برنامه فردی به اجزاء تجزیه می شود.

معماری در بزرگ با معماری سیستم های پیچیده سازمانی که شامل سایر سیستم ها، برنامه ها و اجزای سازنده است، نگران است. این سیستم های سازمانی بر روی رایانه های مختلف توزیع می شود که ممکن است متعلق به شرکت های مختلف باشد.

# مزایای معماری صریح

- معماری ممکن است به عنوان یک مرکز از بحث توسط ذینفعان سیستم استفاده شود
  - تجزیه و تحلیل سیستم
- می تواند تجزیه و تحلیل کند که آیا سیستم می تواند نیازهای غیر کاربردی خود را برآورده کند یا خیر
  - استفاده مجدد در مقیاس بزرگ
- معماری ممکن است در طیف وسیعی از سیستم ها قابل استفاده مجدد باشد
  - معماری های خط تولید ممکن است توسعه یابد

# آنچه معماری ارائه می دهد

- نمودارهای ساده و غیر رسمی که نشان دهنده اشخاص و روابط هستند، اغلب روش مورد استفاده برای مستند سازی معماری های نرم افزاری هستند.
- اما اینها مورد انتقاد قرار گرفته اند؛ زیرا آنها معناشناسی ندارند، انواع روابط بین نهادها و خواص قابل مشاهده نهادهای معماری را نشان نمی دهند.
- بستگی به استفاده از مدل های معماری دارد. الزامات معناشناختی مدل به نحوه استفاده از مدل بستگی دارد.
- بسیار خلاصه (به عنوان مثال، جعبه و نمودار خط) - آنها ماهیت روابط مولفه و خواص خارج از سیستم های زیر را نشان نمی دهد
- با این حال، برای ارتباط با ذینفعان و برنامه ریزی پروژه مفید است.

# استفاده از مدل های معماری

- به عنوان راهی برای تسهیل بحث در مورد طراحی سیستم
  - یک دیدگاه معماری سطح بالا در مورد یک سیستم برای ارتباط با ذینفعان سیستم و برنامه ریزی پروژه مفیدی است، زیرا با جزئیات کامل نیست
    - شرکت کنندگان می توانند با آن ارتباط برقرار کنند و دیدگاه انتزاعی سیستم را درک کنند. سپس آنها می توانند بدون توجه به جزئیات، سیستم را به طور کامل مورد بحث قرار دهند
  - به عنوان راهی برای ثبت یک معماری که طراحی شده است
    - هدف این است که یک مدل کامل سیستم ایجاد کنیم که مولفه



# تصمیمات معماری طراحی

# تصمیمات معماری طراحی

- طراحی معماری یک فرایند خلاقانه است، بنابراین فرایند متفاوت است بسته به نوع سیستم توسعه یافته است
- با این وجود، تعدادی از تصمیمات رایج در تمام فرایندهای طراحی قرار دارند و این تصمیمات ب میزگاههای عملکردی سیستم تاثیر می‌گذارد



# استفاده مجدد از معماری

- سیستم های در همان دامنه اغلب معماری مشابهی دارند.  
که مفاهیم دامنه را بازتاب می دهند.
- خطوط محصول کاربردی در اطراف یک معماری هسته  
ای با انواع مختلفی که نیازهای خاص مشتری را برآورده  
می شود ساخته شده است.
- معماری یک سیستم ممکن است در یکی از الگوهای  
معماری بیشتر یا "سبک" طراحی شود.
- این جوهر یک معماری را جذب می کند و می تواند به  
روش های مختلفی مورد استفاده قرار گیرد.

# معماری و ویژگی های سیستم

- کارایی
  - محلی سازی عملیات بحرانی و به حداقل رساندن ارتباطات از اجزای بزرگ و نهاده های خوب استفاده کنید.
- امنیت
  - از معماری لایه ای با دارایی های بحرانی در لایه های داخلی استفاده کنید.
- ایمنی
  - ویژگی های ایمنی بحرانی را در تعدادی از زیر سیستم ها محلی سازی کنید.
- دسترسی
  - شامل مولفه های بیش از حد و مکانیزم برای تحمل خطا.
- قابلیت نگهداری
  - از اجزای قابل جابجایی استفاده کنید.



# دیدگاه های معماری

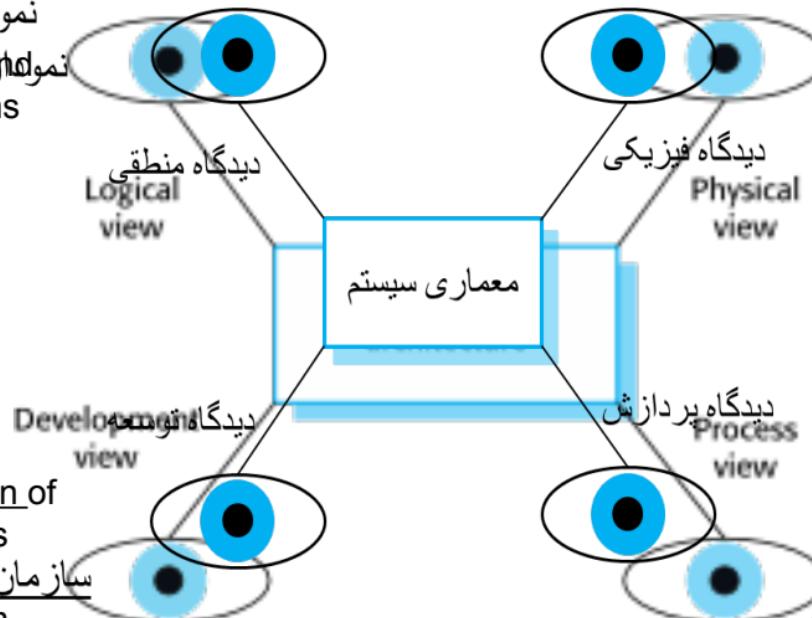
# دیدگاه های معماری

- چه دیدگاه ها و دیدگاه ها هنگام طراحی و مستند سازی یک معماری سیستم مفید هستند؟
- چه توصیفاتی باید برای توصیف مدل های معماری استفاده شود؟
- هر مدل معماری فقط یک دیدگاه یا دیدگاه سیستم را نشان می دهد.
- ممکن است نشان دهد که چگونه یک سیستم به مارژول ها تجزیه می شود، چگونه فرایندهای زمان اجرا تعامل می کنند یا راه های مختلفی که اجزای سیستم در یک شبکه

# دیدگاه های معماری

به طور معمول شامل

نمودارهای کلاس  
typically includes  
class diagrams and  
نمودارهای رفتار  
behavior diagrams



نگاشت (انتقال)

mapping of  
software onto  
hardware

دیدگاه فیزیکی  
Physical view

دیدگاه بردازش  
Process view

view describes  
concurrent  
threads and  
processes

دیدگاه های همزمان  
concurrent  
views  
و فرآیندهای همزمان  
and processes  
را شرح می دهد

static organization of  
the software in its  
development environment  
سازمان استقرار  
در میان توسعه در  
the programmers'  
از دیدگاه برنامه نویسان  
perspective

# مدل 4 + 1 معماری نرم افزار (توسط فیلیپ) (کراجتن)

- یک دیدگاه منطقی که انتزاع کلیدی در سیستم را به عنوان اشیاء یا کلاس‌های شیء نشان می‌دهد.
- یک نمایش فرآیند، که نشان می‌دهد چگونه در زمان اجرا، سیستم متشکل از فرآیندهای تعامل است.
- یک دیدگاه توسعه، نشان می‌دهد که چگونه نرم افزار برای توسعه تجزیه می‌شود.
- نمایش فیزیکی، که نشان می‌دهد ساخت افزار سیستم و نحوه اجزای نرم افزار در میان پردازنده‌های سیستم توزیع شده است
- با استفاده از موارد یا سناریوهای مرتبط (+1) مرتبط است



# الگوهای معماري

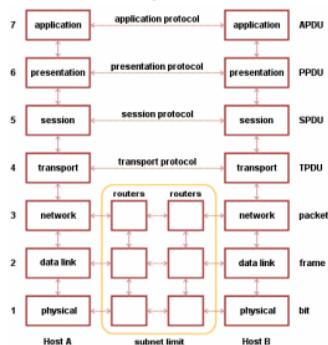


# معماری لایه ای

- برای مدل سازی رابطه های زیر سیستم ها (گروه های زیر کار) استفاده می شود.
  - سیستم را به مجموعه ای از لایه ها (یا ماشین های انتزاعی) که هر یک از آنها مجموعه ای از خدمات را ارائه می دهد، سازماندهی می کند.
- از توسعه تکمیلی سیستم های زیر در لایه های مختلف پشتیبانی می کند. هنگامی که یک رابط لایه تغییر می کند، فقط لایه مجاور تحت تاثیر قرار می گیرد.
- بنابراین، اغلب مصنوعی برای ساختن سیستم ها در این

# معماری لایه ای

- برای مدل سازی ارتباطات زیر سیستم ها (گروه های زیر کار) استفاده می شود.
- سیستم را به مجموعه ای از لایه ها (یا ماشین های انتزاعی) که هر یک از آنها مجموعه ای از خدمات را فراهم می کند، سازماندهی می کند.
- پشتیبانی از توسعه افزایشی زیر سیستم ها در لایه های مختلف. هنگامی که یک رابط لایه تغییر می کند، فقط لایه مجاور تحت تاثیر با این حال، اغلب مصنوعی برای ساخت سیستم در این پشتیبانی از توسعه افزایشی زیر سیستم ها در لایه های مختلف. هنگامی که یک رابط لایه تغییر می کند، فقط لایه مجاور تحت تاثیر با این حال، اغلب مصنوعی برای ساخت سیستم در این پشتیبانی از توسعه افزایشی زیر سیستم ها در لایه های مختلف. هنگامی که یک رابط لایه تغییر می کند، فقط لایه مجاور تحت تاثیر با این حال، اغلب مصنوعی برای ساخت سیستم در این



- استفاده برنامه های کاربردی دسکتاب عمومی
- برنامه های کاربردی وب الکترونیکی تجارت الکترونیک

- دو سبک اصلی
  - لایه بندی شدید
  - لایه بندی آرام

# الگوی معماری لایه‌ای

نام	شرح	مثال	زمان استفاده	مزایا	معایب
معماری لایه‌ای	<p>این سیستم را به لایه‌هایی با قابلیت‌های مرتبط با هر لایه سازماندهی می‌کند. یک لایه خدماتی را به لایه بالای آن ارائه می‌دهد، بنابراین لایه‌های پایین‌ترین سطح، خدمات اصلی را نشان می‌دهند که احتمالاً در سراسر سیستم استفاده می‌شوند. شکل 6.6 را ببینید.</p>	<p>یک مدل لایه‌ای از یک سیستم برای به اشتراک گذاری مدارک حق چاپ در کتابخانه‌های مختلف، همانطور که در شکل 6.7 نشان داده شده است.</p>			

# یک معماری لایه‌ای عمومی

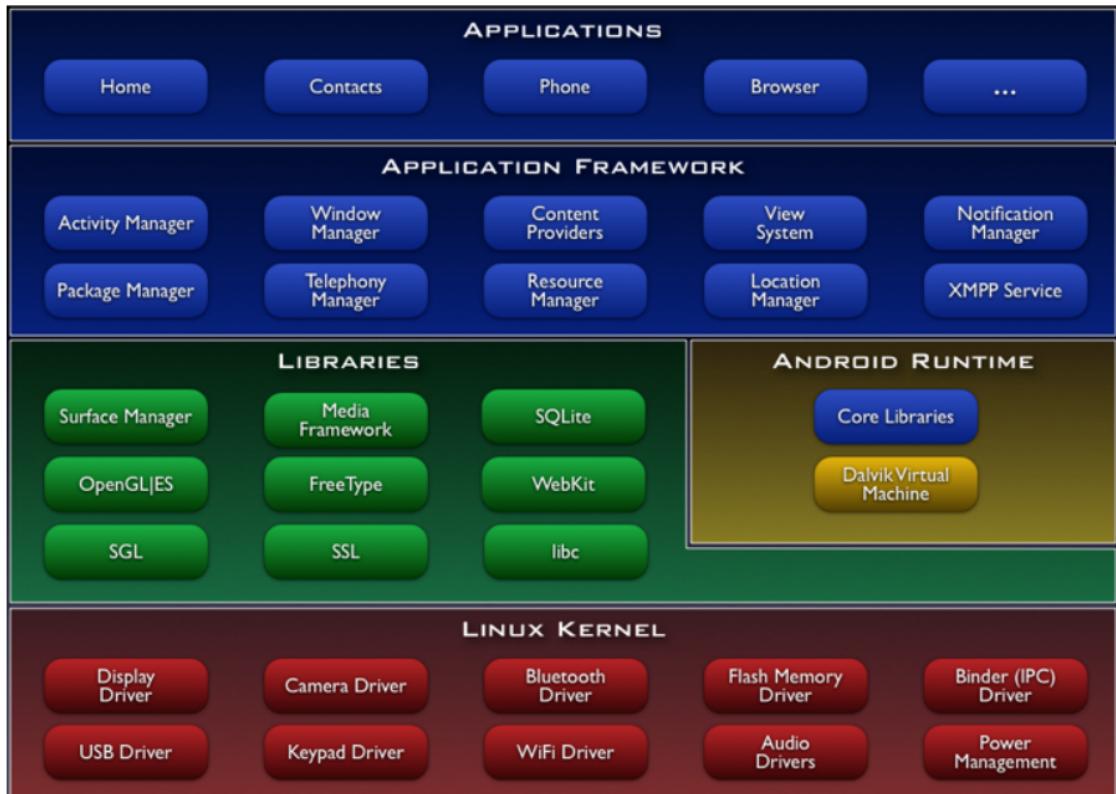
User interface

User interface management  
Authentication and authorization

Core business logic/application functionality  
System utilities

System support (OS, database etc.)

# معماری سیستم آندروید





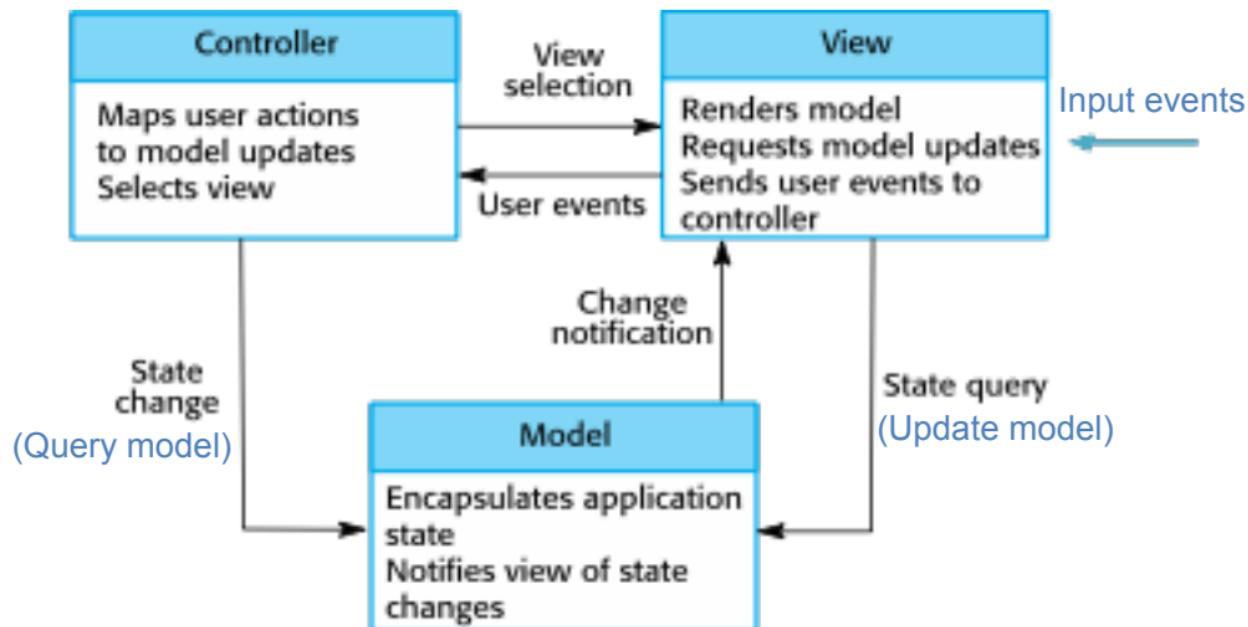
# الگوی Model-View-Controller (MVC)

- شناخته می شود، یک برنامه MVC این الگو، همچنین به عنوان الگوی تعاملی را به ۳ قسمت تقسیم می کند
  - مدل - حاوی قابلیت های اصلی و داده ها است.
  - نمایش - نمایش اطلاعات به کاربر (بیش از یک دیدگاه ممکن است تعریف شده است)
  - کنترل کننده - ورودی را از کاربر کنترل می کند.
- این کار برای جدا کردن بازرس داخلی اطلاعات از روش هایی است که اطلاعات ارائه شده و از کاربر پذیرفته شده است. این قطعات را جدا می کند و اجازه می دهد تا مجددا استفاده از کد مناسب انجام شود.
  - استفاده
  - در زبان های برنامه نویسی عمدۀ WWW معماری برای برنامه های چارچوب وب مانند جنگو و ریل.

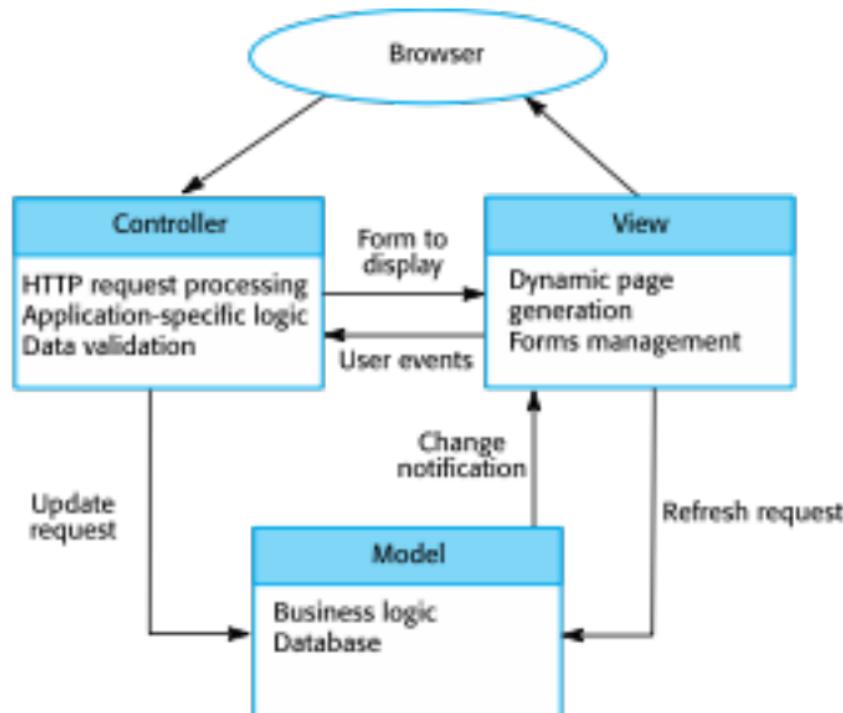
# الگوی Model-View-Controller (MVC)

نام	توصیف
(MVC) نمایشگر کنترل مدل	ارائه و تعامل را از داده های سیستم جدا می کند. این سیستم با سه جزء منطقی که با یکیگر تعامل دارند سازگار است. اجزای مدل اطلاعات سیستم و عملیات مرتبه با آن داده ها را مدیریت می کند. اجزای نمایشگر تعریف و مدیریت داده ها را به کاربر ارائه می دهد. اجزای کنترل کننده تعاملات کاربر را مدیریت می کند (به عنوان مثال، فشار کلید، کلیک ماوس، و غیره) و این تعاملات را به نمایش و مدل منتقل می کند. به شکل 6.3 نگاه کنید.
MVC شکل 6.4 معماری یک سیستم کاربردی مبتنی بر وب که با استفاده از الگوی سازماندهی می کند را، نشان می دهد	مثال
هنگامی که روش های متعددی برای مشاهده و تعامل با داده ها استفاده می شود مورد استفاده قرار می گیرد. همچنین زمانی که نیازهای آینده برای تعامل و ارائه اطلاعات ناشناخته است مورد استفاده قرار می گیرد	موارد کاربرد
به داده ها اجازه می دهد به طور مستقل از خود تغییر کند و برعکس. پشتیبانی از ارائه داده های مشابه به روش های مختلف با تغییرات ساخته شده در یک نمایش نشان داده شده در همه آنها	مزایا
زمانی که مدل داده ها و تعاملات اسیلاند، هیبلنل می تواند بسیار کم و بسیار کم	معایب

# سازمان بندی-view-controller-model



# معماری وب برنامه با استفاده از الگوی MVC



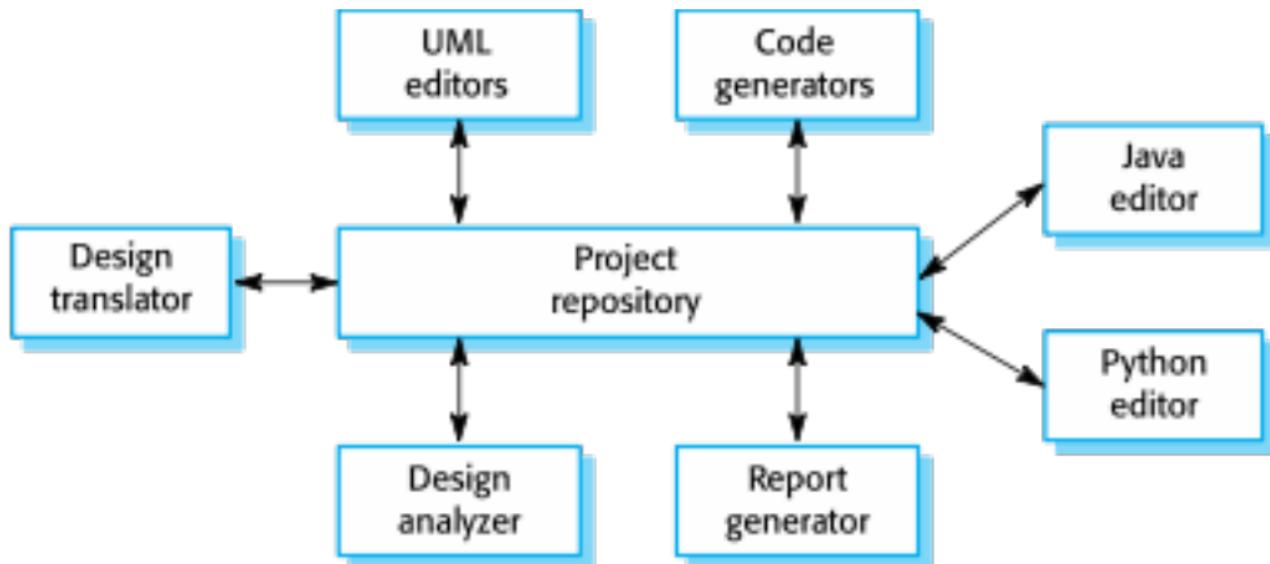
# معماری مخزن

- زیر سیستم ها باید داده ها را مبادله کنند. این ممکن است به دو صورت انجام شود
  - داده های به اشتراک گذاشته شده در یک پایگاه داده مرکزی یا مخزن
  - نگهداری می شوند و ممکن است توسط تمام سیستم های زیر قابل دسترسی باشند؛
- هر زیرسیستم پایگاه داده خود را حفظ می کند و به صورت صریح به سایر سیستم های زیر انتقال می دهد
- هنگامی که مقادیر زیادی از داده ها به اشتراک گذاشته شود، مدل مخزن به اشتراک گذاری رایج ترین استفاده می شود؛ این یک مکانیسم به اشتراک گذاری داده های کارآمد است.

# الگوی مخزن

نام	توصیف
مخزن	تمام اطلاعات در یک سیستم در یک مخزن مرکزی مدیریت می شود که برای تمام اجزای سیستم قابل دسترسی است. کامپوننت ها به طور مستقیم با یکدیگر ارتباط برقرار نمی کنند، فقط از طریق مخزن.
مثال	است که اجزای آن از مخزن اطلاعات طراحی سیستم استفاده IDE شکل 6.9 مثالی از می کنند. هر ابزار نرم افزاری اطلاعاتی را تولید می کند که برای ابزارهای دیگر قابل استفاده است.
زمان استفاده	شما باید از این الگوی زمانی استفاده کنید که یک سیستم داشته باشید که در آن حجم زیادی از اطلاعات ایجاد شده است که باید برای مدت طولانی ذخیره شود. شما همچنین می توانید از آن در سیستم های مبتنی بر داده استفاده کنید که در آن محتویات داده ها در مخزن یک عمل یا ابزار ایجاد می کند.
مزایا	کامپوننت ها می توانند مستقل باشند - آنها نیازی به دانستن وجود اجزای دیگر ندارند. تغییرات ساخته شده توسط یک جزء می تواند به تمام اجزاء منتقل شود. تمام اطلاعات را می توان به طور مداوم مدیریت کرد (به عنوان مثال، پشتیبان گیری انجام شده در همان زمان) به عنوان همه در یک مکان است.

# یک معماری مخزن برای IDE



# معماری مشتری-سرور

- این الگو شامل دو طرف است؛ یک سرور و چندین مشتری
- مدل سیستم توزیع شده که نشان می‌دهد چگونه داده‌ها و پردازش در میان تعدادی از اجزای توزیع شده است
- می‌تواند در یک تک کامپیوتر اجرا شود.
- مجموعه‌ای از سرورهای مستقل که خدمات خاصی نظیر چاپ، مدیریت داده‌ها و غیره ارائه می‌دهند
- مجموعه‌ای از مشتریانی که این خدمات را فراخوانی می‌کنند
- شبکه‌ای که به مشتریان اجازه می‌دهد تا به سرور دسترسی داشته باشند.

باشند

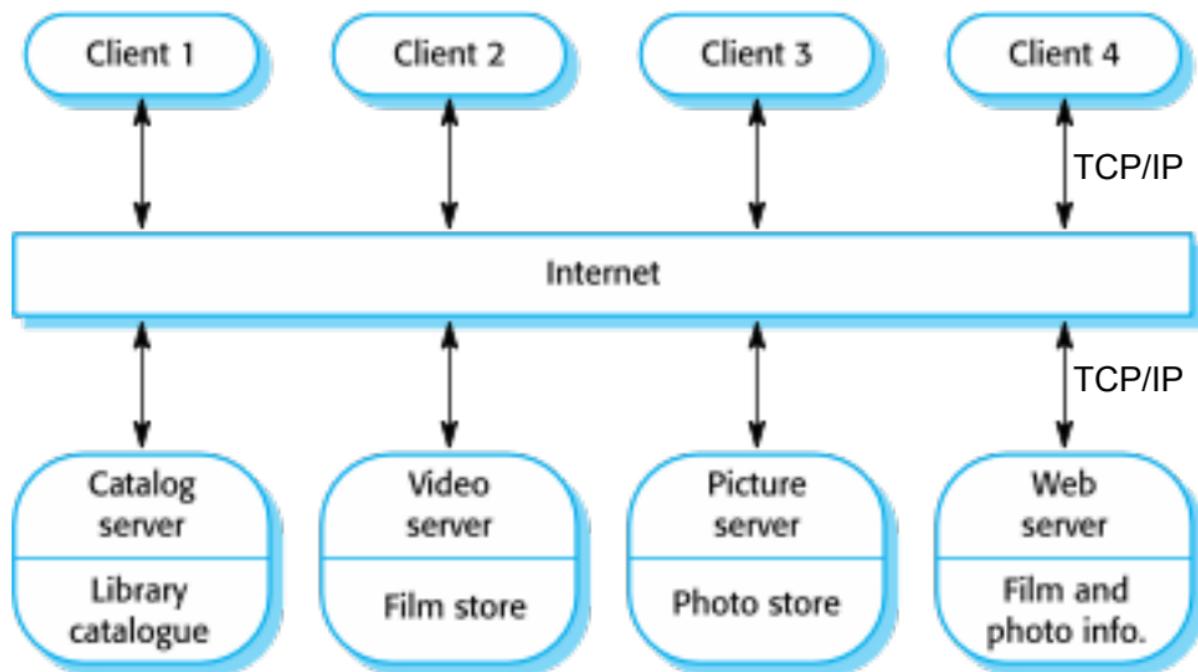
استفاده

• برنامه‌های آنلاین مانند ایمیل، اشتراک سند و بانکداری

# الگوی مشتری-سرور

نام	B
توصیف	در معماری سرویس گیرنده سرور، عملکرد سیستم در سرویس ها سازمان یافته است، هر سرویس از یک سرور جداگانه تحویل می شود. مشتریان از این سرویس ها استفاده می کنند و سرورها را برای استفاده از آنها استفاده می کنند.
مثال	است که به DVD / شکل 6.11 نمونه ای از یک کتابخانه فیلم و ویدئو عنوان یک سیستم سرویس گیرنده سرور سازماندهی شده است.
زمان استفاده	مورد استفاده قرار می گیرد زمانی که داده ها در یک پایگاه داده مشترک باید از طیف وسیعی از مکان ها قابل دسترسی باشند. از آنجا که سرورها می توانند تکرار شوند، ممکن است هنگام بارگذاری بر روی سیستم متغیر باشد.
مزایا	مزیت اصلی این مدل این است که سرورها را می توان در یک شبکه توزیع کرد. قابلیت کلی (به عنوان مثال، سرویس چاپ) میتواند برای همه مشتریان در دسترس باشد و نیازی به اجرای همه سرویسها نباشد.

# معماری مشتری-서버 برای کتابخانه فیلم

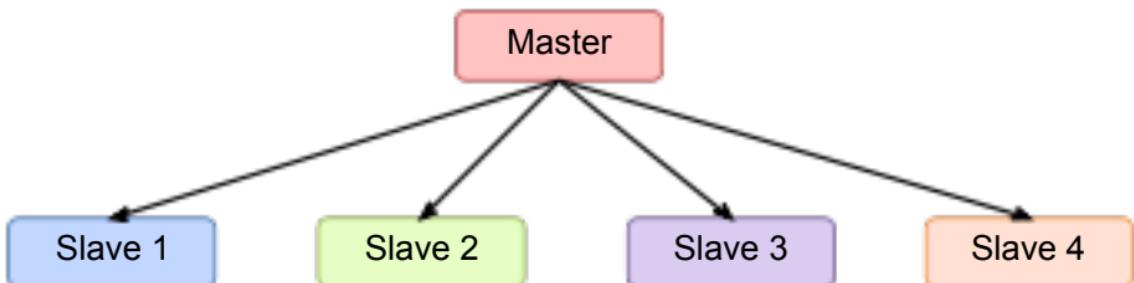


# الگوی ارباب- برده

- این الگو شامل دو طرف است؛ استاد و بردگان مولفه اصلی کار را در میان اجزای یکپارچه برده توزیع می کند و نتیجه نهایی را از نتایجی که بردگان بر می گرداند محاسبه می کنند

- استفاده

- در تکرار پایگاه داده، پایگاه داده اصلی به عنوان منبع معتبر در نظر گرفته می شود، و پایگاه های فرعی به آن هماهنگ می شوند.
- (لوازم جانبی متصل به یک اتوبوس در یک سیستم کامپیوتری (درایوهای اصلی و فرعی).



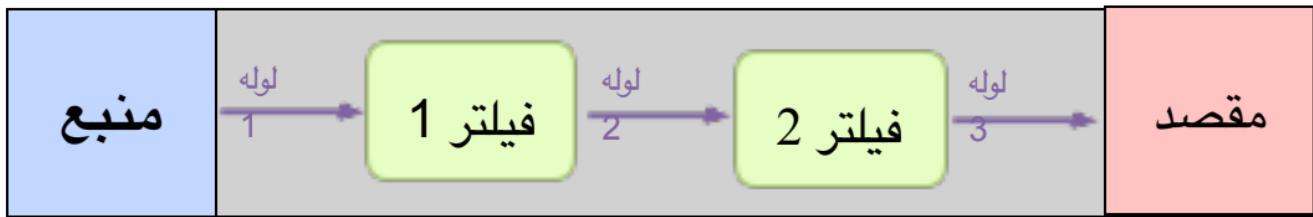
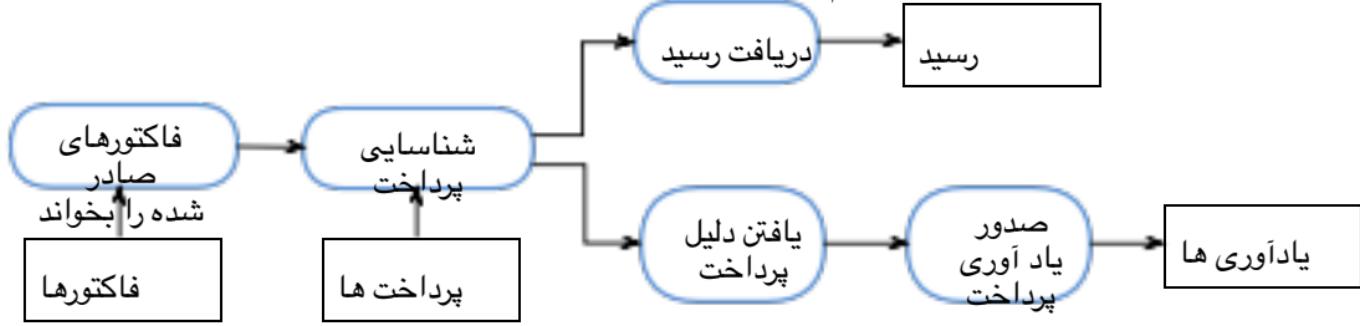
# معماری لوله و فیلتر

- تحولات عملیاتی ورودی های خود را برای تولید خروجی پردازش می کنند.
- گزینه های این روش بسیار رایج هستند. هنگامی که تحولات جریان یا پیوسته است، این یک مدل پیوسته دسته ای است که به طور گسترده در سیستم های پردازش داده استفاده می شود.
- هر مرحله پردازش در یک مولفه فیلتر محصور شده است. داده های مورد پردازش از طریق لوله ها منتقل می شود. این لوله ها می توانند برای بافر یا برای اهداف هماهنگ سازی استفاده شوند.
- استفاده
- کامپایلرها فیلترهای متوالی تجزیه و تحلیل واژگان، تجزیه، تجزیه و تحلیل معنایی و تولید کد را انجام می دهند.
- گردش کار در بیوانفورماتیک
- واقعا برای سیستم های تعاملی مناسب نیست

# الگوی لوله و فیلتر pipe and filter-

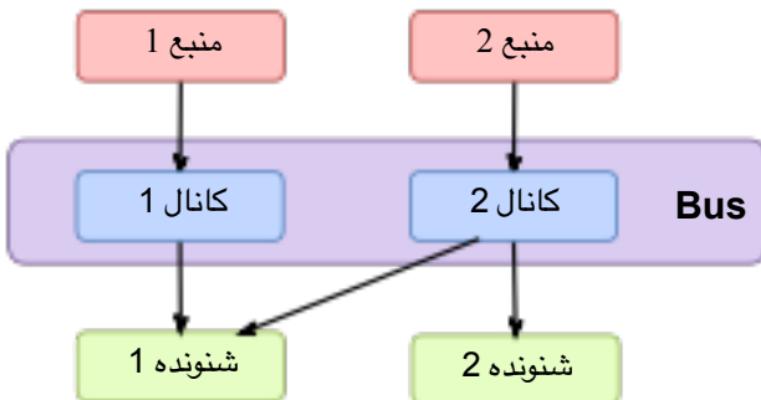
نام	پردازش داده ها در یک سیستم سازماندهی شده است به طوری که هر یک از اجزای پردازش (فیلتر) گستته و یک نوع تحول داده را انجام می دهد. اطلاعات (مانند یک لوله) از یک جزء به جزء دیگر برای پردازش جریان دارد
توصیه	شکل 6.13 نمونه ای از یک سیستم لوله و فیلتر است که برای پردازش صورتحساب استفاده می شود
موقعیت استفاده	معمولًا در برنامه های پردازش داده ها (هر دو مبتنی بر دسته ای و مبادله) مورد استفاده قرار می گیرند که ورودی ها در مراحل جداگانه پردازش می شوند تا خروجی های مرتبط را تولید کنند
مزایا	آسان برای یافک و پشتیبانی از استفاده مجدد از تبدیل. سبک جریان کار متناسب با ساختار بسیاری از فرآیندهای کسب و کار است. تکامل با اضافه کردن تغییرات ساده است. می تواند به عنوان یک سیستم متوالی یا همزمان اجرا شود.
معایب	فرمت انتقال داده ها باید بین تحولات ارتباط برقرار شود. هر تحول باید ورودی خود را تجزیه و تحلیل کند و خروجی آن را تجزیه نشده به فرم تواافق شده رساند. این باعث افزایش سربار سیستم می شود و ممکن است به این معنی باشد که استفاده مجدد از تغییرات عملکردی که از ساختارهای داده ناسازکار استفاده می کنند، امکان پذیر نیست.

# یک نمونه از معماری لوله و فیلتر مورد استفاده در سیستم پرداخت



# الگوی Event-bus

- این الگو در ابتدا با وقایع و چهار جزء اصلی مرتبط است. منبع رویداد. منابع پیام ها را bus رویداد، شنوندگان رویداد، کanal و رویداد منتشر می کند. bus به کanal های خاصی در یک شنوندگان به کanal های خاص مشترک می شوند شنوندگان از پیام هایی که به یک کanal که قبلا آنها مشترک شده اند، اطلاع دارند.

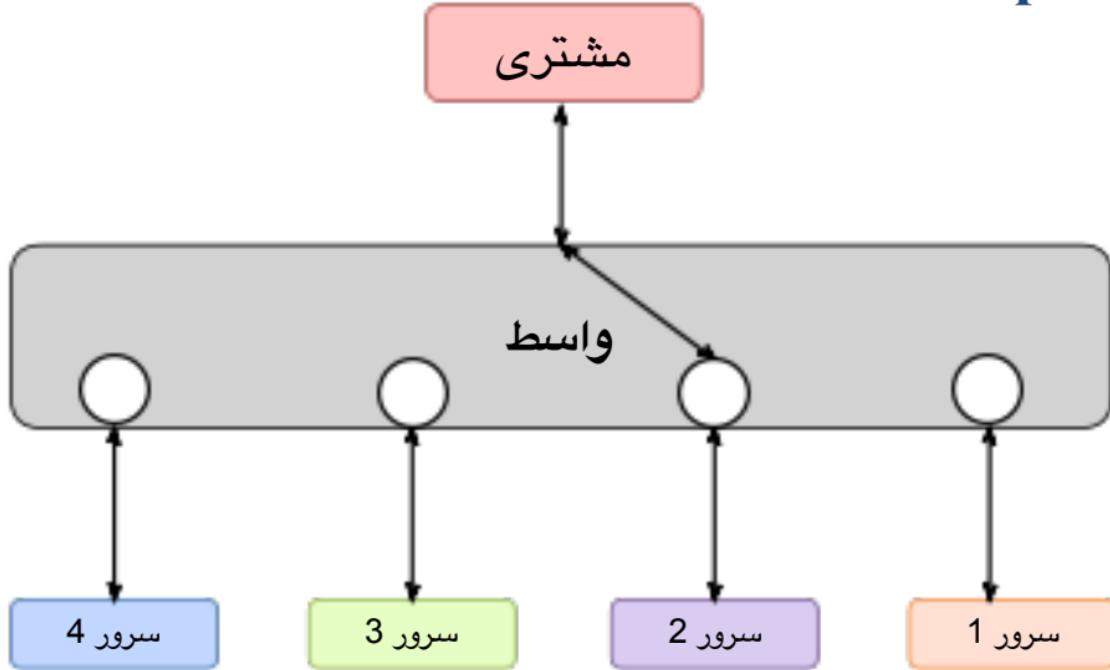


- کاربرد
- توسعه آندروید
- خدمات اطلاع رسانی

# الگوی شرکت واسط (Broker pattern)

- این الگو برای ساختار سیستم‌های توزیع شده با اجزای مجزای استفاده می‌شود . این مولفه‌ها می‌توانند با درخواست از راه دور با یکدیگر تعامل داشته باشند. یک جزء واسط مسئول هماهنگی ارتباط میان اجزا است
- سرورها قابلیت‌های خود ( خدمات و ویژگی‌های ) خود را به یک دلال منتشر می‌کنند . کلاینت یک سرویس از کارگزار درخواست می‌کند ، و کارگزار سپس مشتری را به یک سرویس مناسب از ثبت آن تغییر می‌دهد
- کاربرد
- نرم افزار Apache ActiveMQ ، Apache

# مثالی از الگوی شرکت واسط (broke pattern)

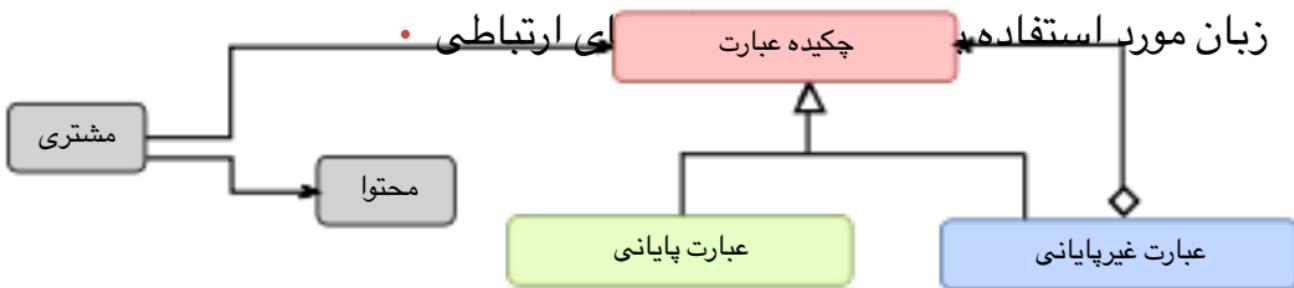


# الگو مترجم (مفسر)

- این الگوی برای طراحی یک جزء است که برنامه های نوشته شده در یک زبان اختصاصی را تفسیر می کند. این به طور عمد نحوه ارزیابی خطوط برنامه ها، به عنوان جملات یا عبارات نوشته شده در یک زبان خاص را تعیین می کند. ایده اصلی این است که یک کلاس برای هر نماد زبان داشته باشیم.

- کاربرد

- SQL زبان جستار پایگاه داده مانند

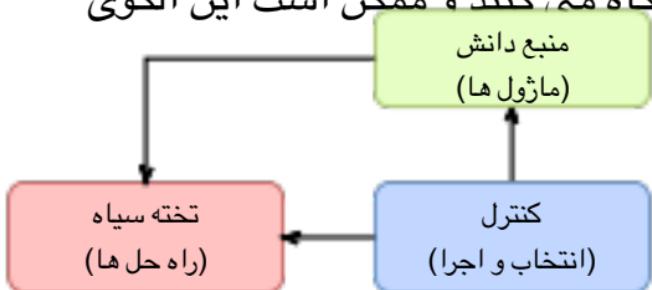


# الگو تخته سیاه

- اين الگو برای مشکلاتی که هیچ استراتژی راه حل غیر قطعی شناخته شده، مفید است و متشکل از ۳ جز اصلی است.

- تخته سیاه - یک حافظه جهانی ساختاری که شامل اجسام از فضای راه حل است.
  - منبع دانش - مأژول‌های تخصصی با نمایندگی خود.
  - جزء کنترل - مأژول‌ها را انتخاب، پیکربندی و اجرا می‌کند.

- همه اجزاء دسترسی به تخته سیاه دارند. قطعات ممکن است اشیاء داده جدیدی را که به تخته سیاه اضافه می‌شوند تولید کنند. کامپوننت‌ها به نوع خاصی از داده‌ها بر روی تخته سیاه نگاه ممکن است این الگوی مطابق با منبع دانش موجود پیدا کنند.

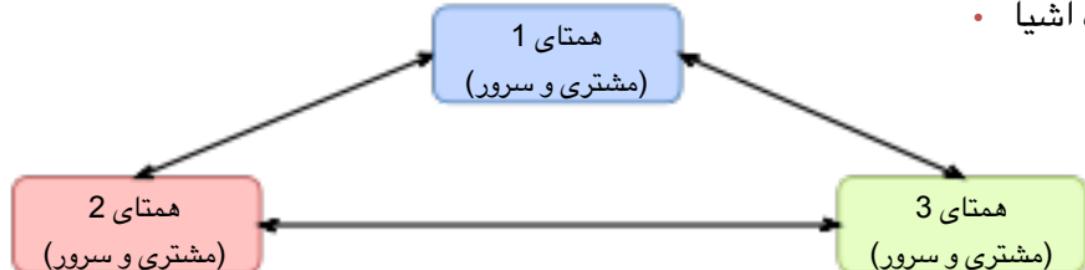


- کاربرد
- تشخیص گفتار
- شناسایی خودرو و ردیابی
- شناسایی ساختار پروتئین

# (الگوی نظیر به همتا به همتا)

- در این الگو ، مولفه های فردی به عنوان **همتا** شناخته می شوند . همتایان ممکن است هم به عنوان یک مشتری عمل کنند ، از همتایان دیگر خود درخواست خدمات کنند ، و به عنوان یک سرور ، خدمات را به همتایان دیگر ارایه کنند . یک همتا به عنوان یک مشتری عمل می کند یا به عنوان یک سرور . یا هر دو عمل می کند و می تواند نقش خود را به طور پویا با زمان تغییر دهد .
  - کاربرد

- Gnutella و G2 شبکه های به اشتراک گذاری فایل مانند
- P2PTV و PDTP پروتکل های چندرسانه ای مانند
- اینترنت اشیا



# مقایسه الگوهای معماری

نام	مزایا	معایب
لایه ای	یک لایه‌ی پایین می‌تواند توسط لایه‌های مختلف دیگری استفاده شود. لایه‌های ساده‌تر می‌شوند زیرا ما می‌توانیم سطوح به وضوح تعریف کنیم. تغییر در لایه بدون تأثیر بر لایه‌های دیگر می‌تواند انجام شود.	جهانی قابل اجرا نیست. ممکن است لایه‌های خاصی در وضعیت خاصی پوشند.
مشتری - سرور	خوب به مدل مجموعه‌ای از خدمات که در آن مشتریان می‌توانند آنها را درخواست کنند	رخواست‌ها معمولاً در موضوعات جداگانه در سرور انجام می‌شود. ارتباطات بین رایانه‌ها موجب سربارگاری می‌شود، زیرا مشتری‌های مختلف دارای نمایندگی های مختلف هستند
ارباب - برد	دقت - اجرای یک سرویس به برد های مختلف، با پیاده سازی های مختلف منتقل می شود	بریدگان جدا شده اند: دولت مشترک نیست آخرین در ارتباطات سپریست می‌تواند یک مسئله باشد، مثلاً در سیستم‌های زمان واقعی این الگوی فقط می‌تواند به یک مشکل که می‌تواند تجزیه شود اعمال می‌شود
لوله - فیلتر	نمایش همزمان پردازش. هنگامی که ورودی و خروجی از جریان ها تشکیل شده است، و فیلتر ها هنگام محاسبه هنگام دریافت داده ها شروع به کار می‌کنند. آسان برای اضافه کردن فیلتر. سیستم را می‌توان به سادگی گسترش داد. فیلترها قابل استفاده مجدد هستند می‌تواند خطوط لوله های مختلف را با ترکیب یک مجموعه داده ای از فیلترها ساخت	کارایی توسعه کمترین فرآیند فیلتر محدود شده است سربرار تغییر داده ها هنگام انتقال از یک فیلتر به دیگری
واسط	اجازه تغییر پویا، اضافه کردن، حذف و انتقال اشیاء را می‌دهد و باعث می‌شود که توزیع به توسعه دهنده شفاف باشد	نیازمند استانداردسازی توصیف های سرویس است
ناظر به ناظر	از محاسبات غیر متربکز پشتیبانی می‌کند. در شکست هر گره داده بسیار قوی است. به شدت مقیاس پذیر از لحاظ منابع و قدرت محاسبات	بیچاره تضمینی در مورد کیفیت خدمات وجود ندارد، زیرا گره ها داوطلب همکاری می‌کنند. امنیت تضمین شده است. عملکرد به تعداد گره ها بستگی دارد
bus - رویداد	ناشران جدید، مشترکین و اتصالات را می‌توان به راحتی اضافه کرد. موثر برای برنامه بسیار توزیع شده است	قیاس پذیری ممکن است یک مشکل باشد، همانطور که همه پیام ها از طریق همان توبوس رویدادی می‌روند
مدل - نمایش - کنترل	امکان مشاهده چندین مدل مشابه را می‌سازد که می‌تواند در زمان اجرا متصل و قطع شود	فزایش پیچیدگی ممکن است به بسیاری از به روز رسانی های غیر ضروری برای اقدامات کاربر منجر شود



# معماری برنامه

# معماری برنامه

- سیستم های کاربردی برای پاسخگویی به نیاز سازمان طراحی شده اند.
- همانطور که کسب و کار ها بسیار مشترک هستند، سیستم های کاربردی خود نیز تمایل دارند که یک معماری مشترک داشته باشند که الزامات برنامه را نشان می دهد.
- یک معماری نرم افزاری معماري یک نوع سیستم نرم افزاری است که ممکن است پیکربندی شده و سازگار با آن برای ایجاد یک سیستم که به شرایط خاص نیاز دارد کاربرد
- به عنوان یک نقطه شروع برای طراحی معماري
- به عنوان یک چک لیست طراحی
- به عنوان راهی برای سازماندهی کار تیم توسعه

# نمونه هایی از انواع برنامه ها

## برنامه های پردازش داده

- برنامه های کاربردی داده ای که پردازش داده ها را در دسته ها بدون دخالت صریح کاربر در هنگام پردازش انجام می دهند.

## برنامه های پردازش تراکنش

- برنامه های متمرکز بر داده که پردازش درخواست های کاربر و به روز رسانی اطلاعات را در یک پایگاه داده سیستم می پردازند. (به عنوان مثال، سیستمهای (تجارت الکترونیک، سیستم رزرواسیون)

## سیستم های پردازش رویداد

- برنامه های کاربردی که فعالیت های سیستم به تفسیر وقایع از محیط سیستم بستگی دارد.

## سیستم های پردازش زبان

- برنامه های کاربردی که در آن نیات کاربران در یک زبان رسمی مشخص شده است که توسط سیستم پردازش شده قابلیت افکار اینها را امتحان می کند.

# سیستم های پردازش تراکنش

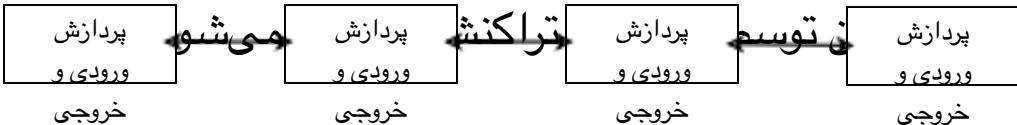
- پردازش درخواست کاربر برای اطلاعات از یک پایگاه داده یا درخواست برای به روز رسانی پایگاه داده.

از دیدگاه کاربر، تراکنش

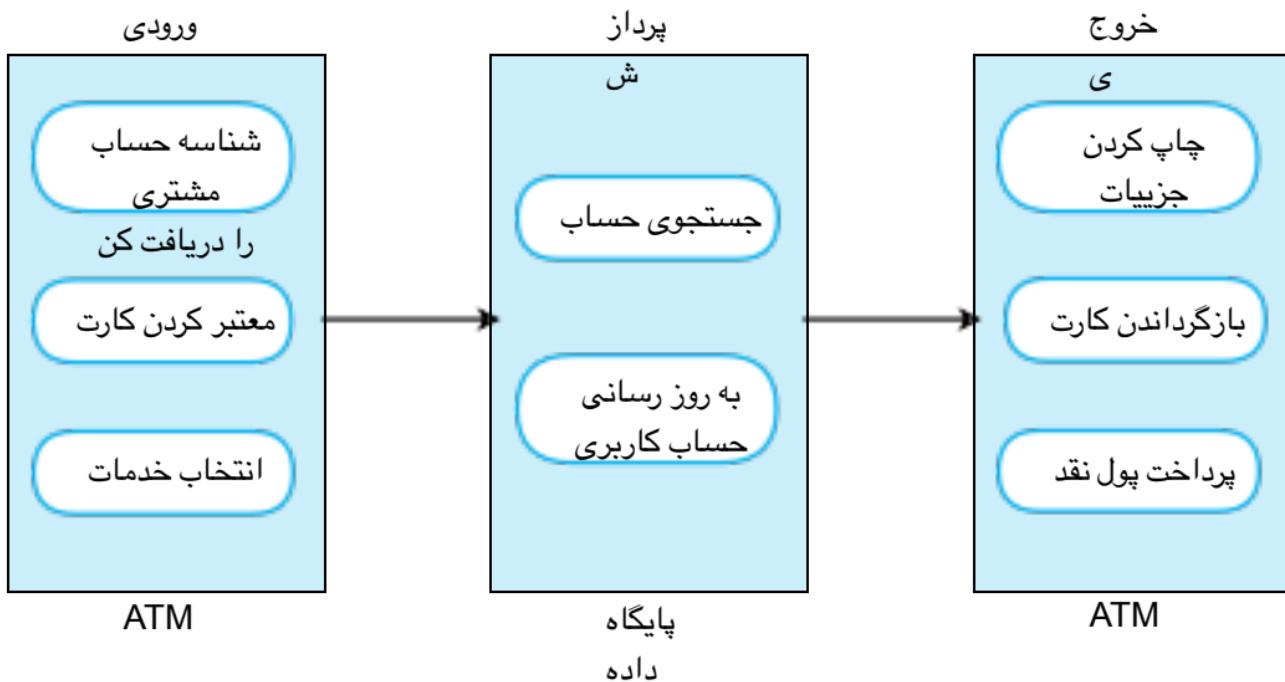
هر دنباله ای منسجم از عملیات که هدف را برأورده می کند، است.

به عنوان مثال - پیدا کردن زمان پرواز از لندن به پاریس

کاربران تقاضای غیر همزمان خدمات را ارایه می دهند که



# معماری نرم افزار سیستم ATM



# معماری سیستم های اطلاعاتی

- سیستم های اطلاعاتی دارای یک معماری کلی هستند که می تواند به عنوان معماری لایه ای سازمان یافته باشد.
- اینها سیستم های مبتنی بر مبادله هستند، زیرا تعامل با این سیستم ها عموماً شامل معاملات پایگاه داده مم شود.

رابط کاربری

ارتباطات کاربر تأیید اعتبار و مجوز

بازیابی اطلاعات و تغییر

پایگاه داده مدیریت معامله

- لایه ها عبارتند از:
  - رابط کاربری
  - ارتباطات کاربر
  - بازیابی اطلاعات
  - پایگاه داده سیستم



# معماری سیستم Mentcare

مرورگر وب

ورود چک کردن نقش فرم و مدیر منو اعتبار سنجی داده ها

مدیریت امنیت مدیر اطلاعات بیمار واردات و صادرات داده تولید گزارش

مدیریت تراکنش  
پایگاه داده بیمار

# سیستم های اطلاعات مبتنی بر وب

سیستم های اطلاعات و مدیریت منابع در حال حاضر معمولاً سیستم های مبتنی بر وب است که در آن رابط کاربر با استفاده از یک مرورگر وب اجرا می شود.

به عنوان مثال، سیستم های تجارت الکترونیک سیستم های مدیریت منابع مبتنی بر اینترنت هستند که سفارشات الکترونیکی را برای کالاهای خدمات می پذیرند و سپس تحويل این کالاهای خدمات را به مشتری ترتیب می دهند.

- در سیستم تجارت الکترونیک، لایه خاص برنامه شامل قابلیت های اضافی پشتیبانی از یک سبد خرید است که در آن کاربران می توانند تعدادی از اقلام را در معاملات جداگانه قرار دهند و سپس آنها را هم در یک معامله واحد برداخت نمایند.<sup>۷</sup>

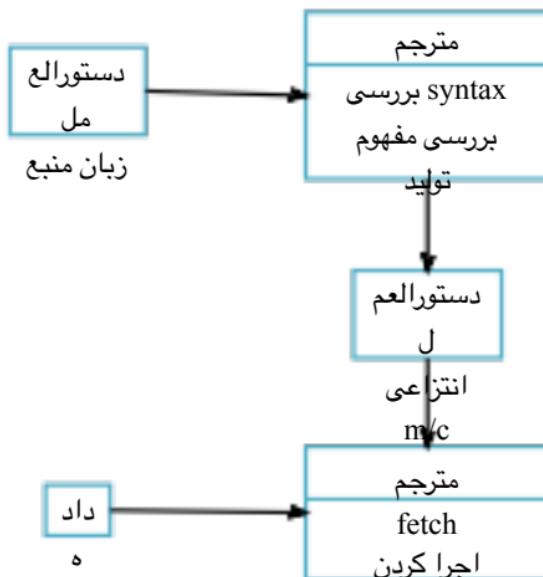
# پیاده سازی سرور

- این سیستم ها اغلب به عنوان یک سرویس دهنده چند هسته ای / معماری (که در فصل 17 مورد بحث قرار می گیرد)

- سرور وب مسئول تمام ارتباطات کاربر است، با استفاده از رابط کاربر با استفاده از یک مرورگر وب؛
- سرور برنامه مسئول اجرای منطق برنامه کاربردی و همچنین درخواست ذخیره سازی و بازیابی اطلاعات است؛
- سرور پایگاه داده اطلاعات را به و از پایگاه داده انتقال می دهد و مدیریت معامله را مدیریت می کند

# سیستم های پردازش زبان

- پذیرش یک زبان طبیعی یا مصنوعی به عنوان ورودی و تولید نمایشی دیگر از آن زبان.
- ممکن است شامل مترجمی باشد که دستورالعمل ها را در زبان

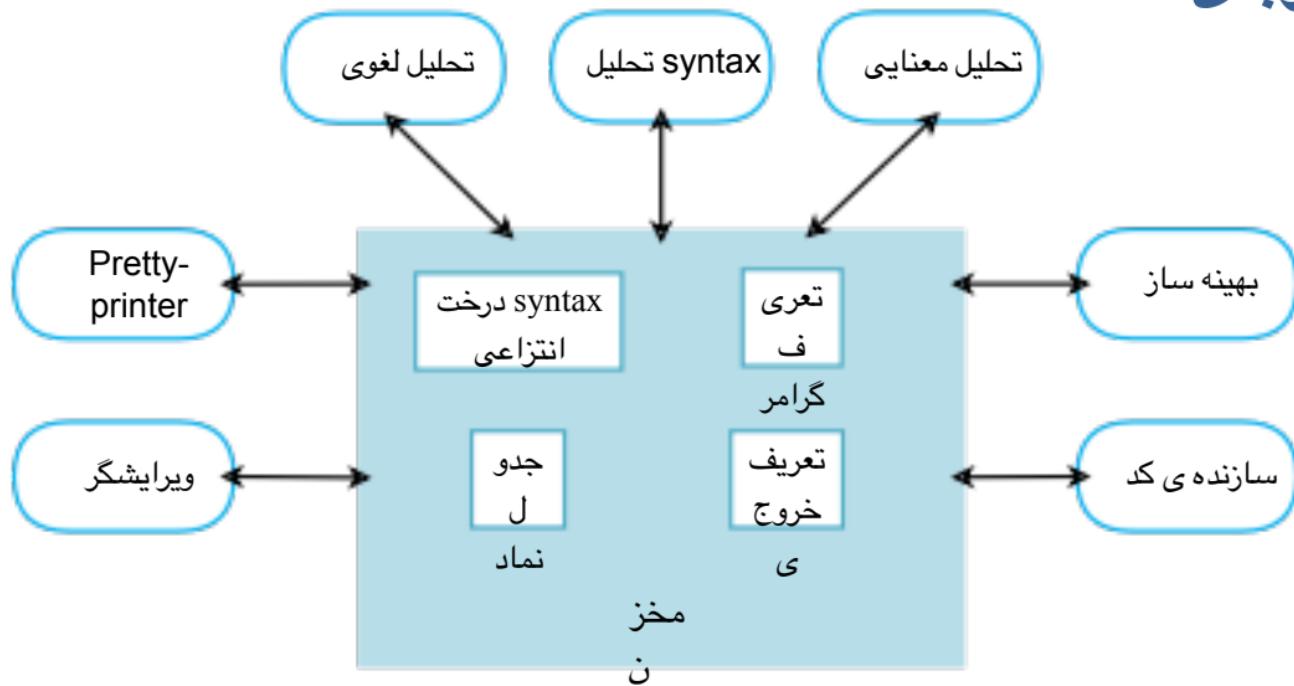


- پردازشی انجام دهد مورد استفاده در شرایطی که ساده ترین راه برای حل یک مشکل، توصیف یک الگوریتم یا توصیف داده های سیستم است توضیحات ابزار فرآیند ابزار.
- متا، قوانین روش، وغیره و تولید ابزار.

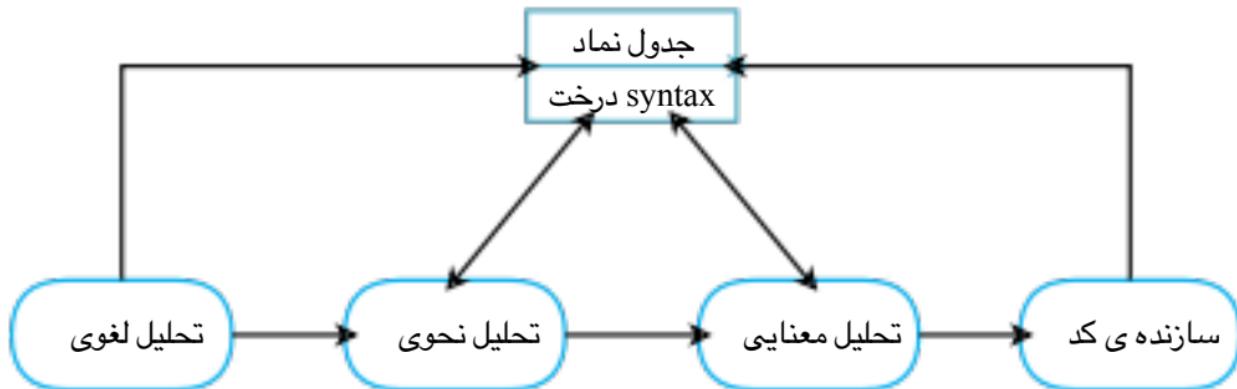
# اجزای کامپایلر

- یک آنالیزگر واژگانی که ورودیهای ورودی را می‌گیرد و آنها را به شکل داخلی تبدیل می‌کند.
- یک جدول نماد که حاوی اطلاعاتی درباره نامها (متغیرها، نام کلاسها، نامهای شی، و غیره) است که در متن ترجمه شده مورد استفاده قرار می‌گیرد.
- یک تجزیه و تحلیل نحو، که نحو زبان ترجمه شده را بررسی می‌کند.
- یک درخت نحوی، که یک ساختار داخلی است که نشان دهنده برنامه کامپایل شده است.
- یک تجزیه کننده معنایی که اطلاعات از درخت نحو و جدول نماد را برای بررسی صحیح معنایی متن متن ورودی استفاده می‌کند.<sup>60</sup>

# یک معماری مخزن برای یک سیستم پردازش زبان



# معماری کامپایلر لوله و فیلتر



# خلاصه

- یک معماری نرم افزاری توصیف یک سیستم نرم افزاری است.
- تصمیمات طراحی معماری شامل تصمیم گیری در مورد نوع برنامه، توزیع سیستم، سبک های معماری مورد استفاده قرار می گیرد.
- معماریها ممکن است از دیدگاه های مختلف مانند یک دیدگاه مفهومی، یک دیدگاه منطقی، یک دیدگاه فرآیند و یک دیدگاه توسعه مستند شوند.
- الگوهای معماری ابزار استفاده مجدد از دانش در مورد معماری سیستم عمومی است. آنها معماری را توصیف می کنند، توضیح می دهند که چگونه ممکن است مورد استفاده قرار گیرد و مزايا و معایب آن را توضیح دهد.
- مدل های معماری سیستم های کاربردی به ما در درک و مقایسه برنامه ها کمک می کنند، سیستم های برنامه کاربردی معتبر را ارزیابی می کنند و اجزای مقیاس بزرگ را برای استفاده مجدد ارزیابی می کنند.
- سیستم های پردازش تراکنش سیستم های تعاملی هستند که اجازه می دهد تا اطلاعات در یک پایگاه داده از راه دور تعدادی از کاربران دسترسی و تعديل شوند.
- سیستم های پردازش زبان برای ترجمه متون از یک زبان به زبان دیگر و اجرای

- **Chapter 6.** I. Sommerville. **Software Engineering.** 10th Edition, Pearson, 2016.
- **Chapter 7,** J. S. Valacich, J. George, **Modern Systems Analysis and Design.** 8th Edition, Pearson 2017.

# با تشکر از دانشجویان دانشگاه علم و فرهنگ که برگردان اسلایدها توسط آنها آماده شده است

- احمدی کیانا
- اسدی امیر حسین
- اسمعیل زاده سالستانی مریم
- باقری فایزه
- بختیاری سهند
- بمانی زاده مینا
- بهرامی ماهان
- تبریزی نرگس
- جدیدی امیرحسین
- جوینده بهزاد
- چنگیزی امیرعلی
- حاجی زین العابدینی
- امیدرضا
- خیرابادی علی
- رحیم پور پیام
- رضائی امیر حسین
- رفیعی سمیرا
- ریاضی امیرعباس
- زیدی مرتضی
- ساجدآبی بگلو زهرا
- ساکی علیرضا
- سلیمی علی
- سلیمی عمار
- سیفی حسین
- شاملوفرد احمد
- شهرابی فراهانی پیام
- صارقی مهدی
- عبد الهی نعیم
- علمشاهی مرضیه
- علی حسینی سلوط احسان
- عینی پارسا
- قاسمی احمد
- قلی زاده بهنام
- محبی کردسفلی معین
- محمودی مهریزی سید محمد
- مختاریان مهسا
- مغانلو حسن
- موتمن حسین
- مهین رضا
- نیک اندیش محمدصادق
- یاسینی مهدی

