

Overview on Composable UI

نگاهی کلی به رابط کاربری افزودنی

حسین خادمیان^۱

^۱ دانشجوی کارشناسی، مهندسی کامپیوتر، دانشگاه شیراز، شیراز، me@hkhademia.ir

۱۳۹۹/۰۴/۱۸

چکیده

در این مقاله، به معرفی روابط کاربری Composable، مزایا و معایب این سیستم، علت چرخش شرکت های بزرگ فناوری به استفاده از این شیوه ساخت روابط کاربری و همچنین نگاه و طریقه آنها در پیاده سازی می پردازیم.

کلمات کلیدی

UI, Composable UI, Flutter, Jetpack Compose, Swift UI, React JS, React Native.

۱ مقدمه

شیوه کنونی تولید روابط کاربری بر مبنای تعریف عناصر صفحه صفحه نمایش به صورت اشیایی شناخته شده و با جداسازی widget های گوناگون مانند Button، ListView، DropBox، CheckBox، ... استوار است.

این شیوه توصیف لایه های کاربری در پلتفرم های گوناگون، جهت پیاده سازی رابط کاربری برنامه ها در سیستم عامل ها دستگاه های مختلف (از جمله کامپیوتر شخصی، تبلت، موبایل، صفحات وب و ...) پیاده سازی می شد.

۲ تکنولوژی های پیشین

جهت پرداختن به مزایا و معایب لایه های کاربری ساختنی و همچنین بررسی تفاوت های آن با شیوه های پیاده سازی کنونی ابتدا باید آشنایی کلی با وضعیت فعلی داشته باشیم. در ادامه به معرفی چند مورد از شیوه های پر استفاده که با آنها کار کرده ام جهت بررسی میپردازم:

۱-۲ HTML در صفحات وب

صفحات وب مجموعه ای از المنت هاست [۲] که هر کدام مشخص کننده قسمتی از ویژگی های صفحه در حال نمایش هستند. صفحات وب از چندین تگ کلی:

۱. <HTML>: این تگ در برگیرنده کل اطلاعات صفحه است

۲. <BODY>: این تگ در برگیرنده کل محتوای قابل نمایش صفحه است

```
1 <html id="rtl">
2 <head>
3 <title>HTML layout using Table (deprecated)</title>
4 <link rel="stylesheet" href="styles.css">
5 <style>
6 </style>
7 </head>
8 <body>
9 <div id="page-wrap">
10 <div id="header" colspan="2">Header</div>
11 <div id="main">
12 <div id="sidebar" rowspan="2">Sidebar</div>
13 <div id="content">
14 <div id="sub-nav">sub menu</div>
15 <div id="article">
16 <form id="login" method="post">
17 Username: <input type="text" name="user" value="" />
18 Password: <input type="password" name="pass" value="" />
19 <input type="submit" name="login" value="Login"/>
20 </form>
21 </div>
22 <div style="clear: both;"></div>
23 </div>
24 <div id="footer" colspan="2">Footer</div>
25 </div>
26 </body>
27 </html>
```

شکل ۱: نمونه کد HTML

۳. <HEAD>: در این بخش اطلاعات صفحه مانند عنوان و استایل صفحه معرفی می شوند

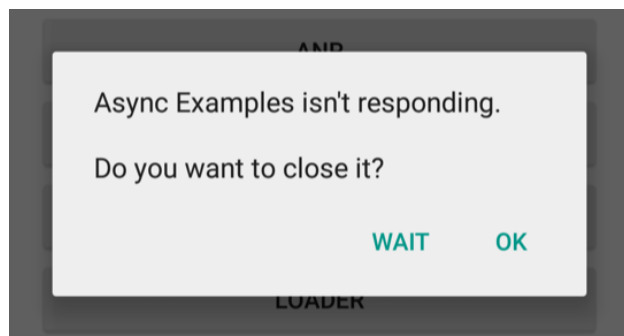
از خواص این نوع تعریف لایه کاربری: تعریف قسمت های مختلف صفحه (دکمه ها، فرم ها، زیرصفحه و ...) در کنار مشخصات و ویژگی های آنها (رنگ، فونت، ...) و حتی کد های رویداد های مختلف در کنار هم است.

چه به صورت مستقیم و متنی و یا با کمک LayoutEditor اقام به تولید این لایه های کاربری می کند. در حین اجرای هر Activity (که نمایانگر قسمتی از منطق برنامه) است، این لایه کاربری بارگزاری می شود و نمایش داده می شود. برای ایجاد هرگونه تغییرات (از جمله افزودن مدیریت رویدادی مثل کلیک) باید ابتدا آن عنصر لایه کاربری را پیدا کرده، رفرنسی از آن را نگه داری و اقدام به اعمال تغییرات مد نظر در آن رفرنس کنیم.[۶]

```
// Imperative style
b.setColor(red)
b.clearChildren()
ViewC c3 = new ViewC(...)
b.add(c3)
```

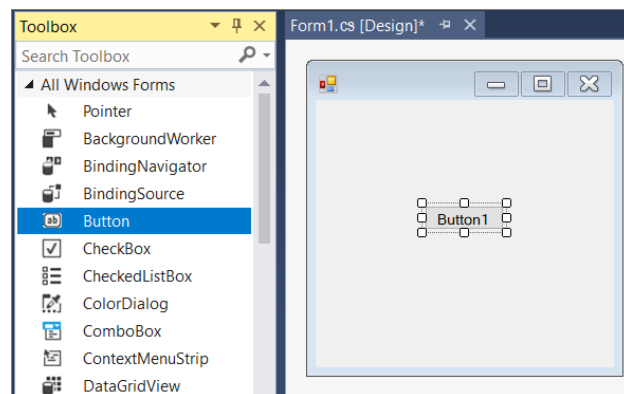
شکل ۳: نمونه کد برنامه Android

لازم به ذکر است که این نگه داری رفرنس در ادامه باعث بروز MemoryLeak ها می شود. این خطای یکی از رایج ترین خطاهای برنامه های اندرویدی است که نیاز به بررسی جداگانه یا توجه مضاعف برنامه نویس جهت مدیریت آنها دارد. [۷] [۸]



شکل ۴: پیام خطایی آشنا، که MemoryLeak یکی از دلایل بروز آن است

سیستم لایه کاربری اندروید، یکی از معیوب ترین سیستم ها در پلتفرم های با بیشترین استفاده را دارد. یکی از این دلایل قدیمی و متفاوت بودن ورژن های مختلف اندروید روی میلیون ها دستگاه فروخته شده در دهه گذشته است. طراحان جهت سازگاری برنامه های تولید شده با ورژن های قدیمی و همچنین معرفی قابلیت های جدید مجبور به پیاده سازی Support Library های مختلف برای قسمت های مختلف محیط کار شدند. به طوری که استفاده از این کتابخانه های پشتیبانی بدون افزودن قابلیت جدید به برنامه ها در اندروید های بروز به حجم فایل نصبی (APK) برنامه چندین مگابایت می افزودند و حتی مصرف رم و حافظه دستگاه افزایش پیدا میکرد. از طرفی استفاده از همین کتابخانه های پشتیبانی در دستگاه های قدیمی تر باعث کاهش سرعت و کارایی و حتی مصرف باتری بیشتر یا داغ کردن دستگاه می شد. در ادامه با معرفی دسته کتابخانه JetPack گوگل تمرکز خود را بر روی ارایه راهکار های خارج از پلتفرم به جای ارایه آنها درون هسته سیستم عامل اقدام کرد. [۹]



شکل ۲: نمونه طراحی با ویرایشگر WinForm

مزایا: [۴]

۱. این مدل توصیفی است، یعنی برنامه نویسی بدون نیاز به کد نویسی به توصیف لایه کاربری می پردازد.
۲. زبان ساده و خوانا، قابل استفاده برای نا-برنامه نویسان (طراحان، دانش آموزان و ...) [۱]
۳. چشم پوشی از خطا: اگر در توصیف لایه کاربری خطایی (در بارگزاری یا حتی سمت برنامه نویسی) رخ دهد کل برنامه بی استفاده نمی شود.

معایب: [۳]

۱. فقط امکان تولید صفحات استاتیک به تنهایی دارد
۲. مخلوطی از تمامی منطق برنامه، ظاهر کاربری، کنترل و اطلاعات. باعث عدم امکان توسعه خطی در پروژه های بزرگ و سازمانی می شود.
۳. نیازمند دیگر ابزار جهت پویا سازی مثل جاوا اسکریپت، activeX و ...
۴. صاحب و مدیر StateMachine است و برنامه نویس فقط با event ها از تغییرات وضعیت با خبر می شود. و کنترل کاملی روی آن ندارد. این مسئله باعث به وقوع پیوستن States nonDeterministic می شود.

۲-۲ WindowsForms, UWP, WPF

پلتفرم WindowsForms تکنولوژی قدیمی بر بستر WindowsAPI ارایه می شود و به طور کلی ابزاری جهت پیاده سازی رابط کاربری گرافیکی برای برنامه های ویندوزی (GUI) است. در این پلتفرم جهت توصیف لایه کاربری مستقیماً با LayoutEditor نسبت به تعریف رابط کاربری اقدام می کنیم. ایده پشت این پلتفرم Form-Based بودن است. به این شیوه که تمامی صفحاتی که کاربر با آن تعامل می کند در حقیقت فرمی جهت ورود اطلاعات و دریافت بازخورد از اوست. نسخه بروزتر از این پلتفرم UWP و WPF که به ترتیب برای توسعه برنامه های چند سکویی و دسکتاپ طراحی شده اند دارای زبانی جهت توصیف لایه کاربری به نام XAML هستند. این شیوه شباهت بسیاری به صفحات HTML داشته و بیشتر مزایا و معایب آن در این شیوه هم صادق است. [۵]

۳-۲ Android

لایه های کاربری در اندروید، به صورت از پیش تعریف شده در فایل XML هایی به عنوان layout برنامه ایجاد می شوند. برنامه نویس امکان ویرایش این فایل را

همچنین فرایند پیچیده تر دریافت گزارش های خطا از کاربر می شود. حتی گاهی مدیریت یک وضعیت ناخواسته از خود منطق برنامه پیچیدگی بیشتری پیدا میکند. یکی دیگر از موارد مهم، Reusability کد ها و کامپوننت هایی که یک برنامه نویس تولید و استفاده می کند است. در شیوه سنتی به دلیل وابستگی کامل برنامه نویس به زمینه ای که در آن یک کامپوننت نوشته می شود (از جمله موقعیت قرار گیری در صفحه، المنت های دیگر صفحه و ...) امکان استفاده از یک کامپوننت در دیگر جاهای برنامه کمتر و فرایند آماده سازی یک کامپوننت برای استفاده در پروژه های دیگر خیلی هزینه زیادی دارد.

اکثر مشکلات اشاره شده در این قسمت و قسمت های قبل از عدم کنترل برنامه نویس بر وضعیت کنونی برنامه و همچنین متمرکز نبودن مرجعیت درستی وضعیت در کل برنامه است. برای مثال: به این صورت که از دید برنامه نویس چک باکس شماره ۱ فعال و چک باکس ۲ غیر فعال است. با تغییر این انتخاب توسط کاربر در اکثر پلتفرم ها ابتدا وضعیت تغییر پیدا کرده و سپس تغییر جدید به برنامه نویس از طریق Event ها اطلاع داده می شود. و برنامه نویس کنترلی روی این تغییر ندارد.

۴ راه کار

راه حل این مسئله فرستادن کل فرایند رسیدگی و مدیریت وضعیت ها به برنامه نویس و رندرینگ و ... به پلتفرم است. همچنین کامپوننت های تشکیل دهنده لایه کاربری هیچ گونه وضعیتی آشکار (قابل مدیریت توسط برنامه نویس) نگه داری نکنند. به این ترتیب حتی تغییر وضعیت یک دکمه رادیویی، تایپ در یک تکست باکس هم با اطلاع و موافقت برنامه نویس صورت میگیرد.

برای پیاده سازی این شیوه تولید کامپوننت های لایه کاربری نیاز است که تغییری در نحوه نگاه به لایه های کاربری به طور کلی بیاندازیم. از جمله:

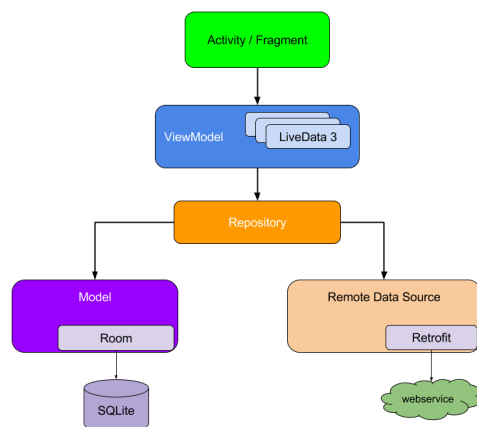
۱. هر کامپوننت فقط باید به نقش انحصاری خود بپردازد و در کار بقیه قسمت ها تغییری ایجاد نکند.
۲. اطلاعات مورد نیاز جهت پیاده سازی به کامپوننت به آن داده می شود. همچنین کامپوننت از هرگونه تغییری در اطلاعات باید آگاه شود.
۳. کامپوننت ممکن است جهت اجرای نقش مشخص شده خود، دارای وضعیتی داخلی باشد که البته هیچ تاثیری در وضعیت کلی برنامه نباید بگذارد.
۴. کامپوننت ها نباید وابستگی به والد یا فرزندان خود داشته باشند و به تنهایی و در هر محیط و usecase بتوانند وظیفه مربوطه را انجام دهند.

یکی از دیدگاه های مفید در این زمینه به این صورت مطرح می شود که:

$$UI = f(state)$$

The layout on the screen Your build methods The application state

شکل ۷: UI is result of application function on state

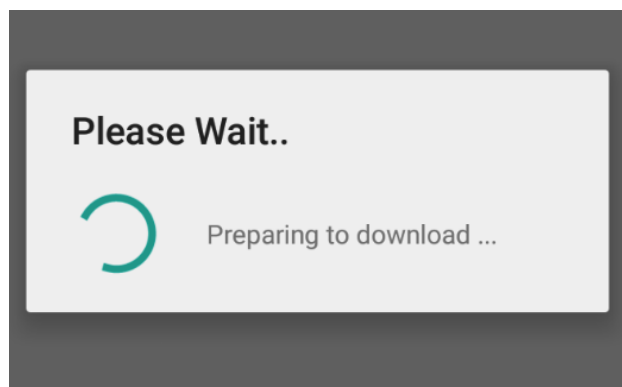


شکل ۵: شمایی از ساختار جدید ارایه شده گوگل برای برنامه ها

۳ زمینه

تا کنون با تعدادی از شیوه های تولید رابط کاربری قدیمی آشنا شدیم. در اکثر این شیوه ها برنامه نویس هیچ گونه مدیریتی بر وضعیت ماشین (برنامه) و محتوای نمایش داده شده به کاربر به طور مستقیم ندارد. بلکه با استفاده از ویجت های ارایه شده هر پلتفرم و با اتکا به رویداد های گزارش شده آنها به بروزرسانی وضعیت مد نظر می پردازد.

این شیوه از مدیریت وضعیت و اتکا به پلتفرم های گوناگون (با امکان آپدیت و تغییرات) باعث مشکلات بسیاری در برنامه ها می شود.



شکل ۶: نمونه دیالوگ درحال بارگذاری قبل از چرخش

اکثر ما تا کنون با برنامه هایی که برخلاف انتظار برنامه نویس به وضعیتی نامطلوب رفته اند (فرضا یک دیالوگ "درحال بارگزاری" در یک برنامه اندرویدی که با چرخش صفحه ناپدید می شود اما از دید برنامه هنوز باید نمایش داده می شد). یکی از مهمترین دلایل این تجارب، نه از عدم آگاهی توسعه دهنده بلکه از **غیر قابل پیش بینی بودن پلتفرم** ایجاد می شود.

جهت مدیریت اینگونه وضعیت های از قبل پیشبینی نشده که اکثرا به علت نقص در پلتفرم، بروز نبودن پلتفرم در دستگاه کاربر (مثل نسخه اندرویدی دستگاه)، عدم شناخت کافی توسعه دهنده از ویژگی ها و عکس العمل های هر کامپوننت در شرایط گوناگون نشأت می گیرد، مستلزم به بکارگیری تیم آزمایش محصول و



شکل ۹: زیرنویس نمودارها باید کامل و جامع باشد، به عبارت دیگر خواننده بتواند با دیدن نمودار و خواندن زیرنویس آن پی به تمامی اطلاعات نهفته در نمودار ببرد.

وی است. این حالت اخیر از دهه آخر قرن بیستم تا کنون به وفور در مقالات معتبر پی گیری شده است تا جایی که در صورت تایید نتایج با شبیه سازی، آن را به نوعی اثبات برای طرح پیشنهادی در نظر می گیرند.

برای آوردن اشکال در کنار یکدیگر می توانید از محیط subfigure استفاده کنید. در شبیه سازی ها می بایست نویسندگان به صورت دقیق و صریح پیکربندی شبیه سازی و مجموعه داده ای که مورد استفاده قرار داده اند را با ذکر منابع و مراجع مورد نیاز، ذکر کنند. در ضمن پارامترهای مورد استفاده برای شبیه سازی باید در همین بخش شبیه سازی تعریف شود. به عنوان مثال اگر شما از پارامتر MSE (mean squared error) استفاده می کنید باید آن را در این بخش تعریف کنید. شکل های شبیه سازی باید واضح و مشخص باشد. دقت کنید به دلیل این که در نهایت مقالات پذیرفته شده به صورت سیاه و سفید چاپ خواهد شد، بدین سان از تمایزگذاری با رنگ های مختلف بین نمودارها کافی نخواهد بود. یک نمونه از تمایزگذاری مناسب را می توانید در نمودار ۱۰ و ۱۱ ملاحظه کنید.

تمامی نمودارهای قسمت شبیه سازی باید دارای Legend باشند. محورهای نمودارها همگی باید دارای برچسب مناسب و همچنین شماره گذاری مناسب باشد. در $LaTeX$ سعی کنید نمودارهای خود را با کیفیت pdf وارد کنید و از قراردادن نمودارهای با کیفیت jpg و یا کیفیت تصویری خودداری کنید. بسیاری از ابزارهای شبیه سازی به شما خروجی pdf را می دهند.

استفاده از نمودارهای نوین به مانند Box-Plot به شدت مورد استقبال قرار می گیرد. نکته مهم در شبیه سازی انجام پذیرفته در مقاله این است که نویسندگان تنها به گزارش میانگین خطا بسنده کرده اند. از دیدگاه علم شبیه سازی این کار نمی تواند اطمینان خواننده را به شبیه سازی جلب کند. به عنوان مثال دو کلاس را در نظر بگیرید که نمرات سه دانشجوی آن ۹، ۱۰ و ۱۱ است و کلاس دیگر ۵، ۱۰ و ۱۵. هر دو کلاس میانگین یکسانی دارند، اما واقعا بین نمرات این دو کلاس تفاوت زیادی وجود دارد. به عنوان پیشنهاد می توانید از نمودار Box-Plot استفاده کنید و یا حداقل بازه اطمینان نتایج را گزارش کنید. نمونه ای از یک Box-Plot در

به این معنی که هرگونه تغییر در وضعیت برنامه در ظاهر کاربری اثر میگذارد و نه برعکس. ظاهر کاربری هیچگونه مسیولیتی در زمینه صحت سنجی، بروز رسانی، مدیریت، بازگذاری و پردازش اطلاعات ورودی و خروجی و همچنین وضعیت فعلی برنامه ندارد و تنها مسئولیت آن به نمایش گذاشتن وضعیت فعلی اطلاعات برنامه است.

۵ پلتفرم های امروزی

در قسمت قبل به بررسی راه حل کلی این مدل پیاده سازی آشنا شدیم. حال در ادامه چندین پیاده سازی این راه کار را بررسی می کنیم:

۱-۵ React

MIT License : Facebook Maintainer این فریم ورک در اصل یک موتور نمایش اطلاعات در DOM است. به همین علت هرگونه مدیریت وضعیت و ... نیازمند کتابخانه های کمکی جهت مدیریت وضعیت برنامه از جمله Redux نیازمند است. کامپوننت های تولید شده توسط این فریم ورک به صورت تگ های HTML درون برنامه استفاده می شوند. همچنین تمامی وضعیت مورد نیاز کامپوننت در ابتدای ایجاد آن کامپوننت به آن ارسال می شود.

```
1 class App extends React.Component {
2   render() {
3     return (
4       <div>
5         <p>Header</p>
6         <p>Content</p>
7         <p>Footer</p>
8       </div>
9     );
10  }
11 }
```

شکل ۸: A simple application component in ReactJS

۱-۱-۵ ویژگی های بخش تحلیل و ارزیابی

در صورتی که روش پیشنهادی حسی نبوده و مبتنی بر استدلال ریاضیاتی باشد، می توان به ارزیابی عملکرد آن در سامانه و نحوه بهبود نتایج نسبت به کارهای گذشته به طور کمی و کیفی پرداخت. بدیهی است اگر مدل پیشنهادی حسی بوده و قابل استناد ریاضیاتی نباشد، تنها می توان به صورت کیفی به ارزیابی عملکرد پرداخت.

۲-۵ ویژگی های بخش شبیه سازی

از آنجایی که بسیاری از تحقیق ها به منظور حل مساله ای عملی پی گیری می شوند، این بخش از اهمیت خاصی برخوردار است. در این بخش به معرفی شبیه سازی صورت گرفته و ارائه نتایج به صورت مطلوب (جدول، نمودار و ...) پرداخته می شود. در مواردی که طرح پیشنهادی اثبات شده است، نتایج باید با تقریب خوبی با مدعا یکسان باشد. در مواردی که طرح پیشنهادی اثبات نشده و به طور حسی و مبتنی بر برخی پیش فرض ها مطرح شده است، اهمیت این بخش بیشتر از حالت قبل است، چرا که نتایج شبیه سازی تنها مستند نویسندگان و به نوعی موید مدعای ثابت نشده



(ج)

(ب)

(آ)

شکل ۱۰: (آ) این شکل در این مورد سخن می‌گوید که، (ب) در مورد این شکل اکنون صحبت کنید....، (ج) این شکل برای

شکل ۱۱ نشان داده شده است.

که شما از فروشگاه iTunes دریافت می‌کنید توسط قالب DRM اپل محافظت می‌شود. این قالب یک قالب فایل AAC انحصاری و محافظت شده است، و اپل مجوز استفاده از آن را به هیچ کس نمی‌دهد.

جداسازی اجزای مختلف یک جمله نیز نقش زیادی در فهم آسان آن دارد. ویرگول میتواند اجزای یک جمله را در جایی که نیاز به مکث هست، از هم جدا کند؛ حال آن که نقطه ویرگول برای جداسازی دوجمله که با هم ارتباط معنایی دارند، بکار میرود. نقطه نیز برای جدا کردن جملات مورد استفاده قرار میگیرد. درکاربرد هالالین (پراتز) باید توجه شود که عبارت داخل آن برای توضیحی است که از اجزای جمله محسوب نشده و درصورت حذف خللی به آن وارد نمیشود. درمقابل، گیومه برای برجسته کردن جزیی از جمله بکار میرود.

تا جای ممکن از بکار بردن کلماتی مثل «میباشد»، «گردید»، و «بوده باشد» پرهیز شود. به جای آن‌ها اغلب می‌توان از کلمات ساده و روان مثل «است» و «شد» استفاده کرد. بکارگیری کلمات دشوار و غیرمعمول تنها باعث پیچیده شدن جمله و دشوار شدن فهم آن می‌شود.

برای کلمات فنی تا حد امکان از معادل‌های پارسی استفاده شود. بدون تردید کلمه «پردازش» زیباتر از «پروسس» است، و یا کلمه «ریزپردازنده» از «میکروپروسسور» مناسب‌تر است. در چنین مواقعی اگر احتمال می‌دهید خواننده با معادل پارسی آشنا نیست، از پانویس برای نوشتن معادل انگلیسی استفاده کنید. این کار را در اولین کاربرد معادل‌های پارسی انجام دهید. مثل گره راهنما^۱ تا حد امکان از کلمات انگلیسی در جملات استفاده نکنید. مثلاً بجای نوشتن Microsoft میتوانید بنویسید: «میکروسافت». اگر ناچار شدید در یک جمله از کلمات انگلیسی استفاده کنید، حتماً فاصله کافی بین آن‌ها و کلمات پارسی راعایت کنید.

۱-۶ علامت‌گذاری

برای خوانایی بهتر مقاله باید سعی شود تا حد امکان علامت‌گذاری متن مقاله بدرستی انجام شود. دقت کنید تمام علامت‌هایی مثل نقطه، ویرگول، نقطه ویرگول، دونقطه، و علامت سوال باید به کلمه قبل از خود چسبیده باشند، و از کلمه بعدی تنها به اندازه یک فضای خالی فاصله داشته باشند. علامت خط تیره باید به اندازه یک فضای خالی از کلمه قبل و بعد از خود فاصله داشته باشد؛ مگر این که کلمه

سعی کنید از قرار دادن کدهای شبیه‌سازی چه در قسمت شبیه‌سازی و چه در قسمت روش پیشنهادی به شدت خودداری کنید. نویسندگان به صورت اختیاری می‌توانند کدهای شبیه‌سازی خود را در یک وبسایت قرار داده و به آن در مقاله پیوند دهند.

۳-۵ ویژگی‌های بخش نتیجه‌گیری

در بخش نتیجه، نکات مهم انجام شده در کار بصورت خلاصه مرور و نتایج به دست آمده توضیح داده شوند. همچنین در این بخش باید سهم علمی مقاله بصورت واضح بیان شود. هرگز عین مطالب چکیده را در این بخش تکرار نکنید. نتیجه می‌تواند به کاربردهای پژوهش انجام شده اشاره کند؛ نکات مبهم و قابل پژوهش جدید را مطرح کند؛ ویا گسترش موضوع بحث را به زمینه‌های دیگر پیشنهاد دهد.

۴-۵ ویژگی‌های بخش پیوست‌ها

بخش پیوست‌ها یک بخش اختیاری است و شماره‌گذاری نمی‌شود. موضوع‌های مرتبط با مقاله که در یکی از گروه‌های زیر قرار گیرند، می‌توانند در بخش ضمایم آورده شوند.

- اثبات ریاضی فرمول‌ها یا الگوریتم‌ها.
- داده‌ها و اطلاعات مربوط به مطالعه موردی.
- نتایج کار دیگر محققان و داده‌های مربوط به مقایسه آن‌ها.
- سایر موضوع‌های مرتبط که جزء بخش‌های اصلی مقاله نباشند.

۶ قواعد نگارشی

شیوایی و رسایی نوشتار در گرو ساده‌نویسی است. تلاش شود در متن مقاله از جملات رسا، گویا، و کوتاه استفاده شود و از نوشتن جملات تودرتو پرهیز شود. به این جمله دقت کنید: «آهنگی که شما از فروشگاه iTunes دریافت می‌کنید توسط قالب DRM اپل که یک قالب فایل AAC انحصاری و محافظت شده است که اپل مجوز استفاده از آن را به هیچ کس نمی‌دهد، محافظت می‌شود». این جمله در واقع از سبک نگارش زبان انگلیسی پیروی می‌کند و به هیچ وجه برای جملات پارسی مناسب نیست. به راحتی می‌توان این جمله را به این صورت بازنویسی کرد: «آهنگی

قبلی یا بعدی يك عدد باشد، که در این صورت باید به آن بچسبد. بین کلماتی که جدا هستند باید يك فضای خالی فاصله باشد.

۲-۶ املا

درستی نوشتار بر پایه املاي زبان پارسی ضروری است. در این بخش برخی از موارد اشتباه متداول را یادآوری می‌کنیم. می‌توانید اطلاعات دقیق‌تر را با مراجعه به کتابهای نوشته شده در این زمینه پیدا کنید.

در افعال حال و گذشته استمراری باید دقت شود که «می» از جزء بعدی فعل جدا نماند. برای این منظور می‌توانید از نیم‌فاصله استفاده کنید.

در مورد «ها» ی جمع نیز دقت کنید که از کلمه جمع بسته شده جدا نوشته شود؛ مگر در کلمات تک هجایی مثل «آنها». برای جدانویسی نیز از فاصله متصل استفاده کنید. مثلاً «پردازنده‌ها» را بصورت «پردازنده‌ها» بنویسید.

جمع بستن کلمات پارسی یا لاتین با قواعد زبان عربی اشتباه است. بنابراین «پیشنه‌ادات» و «اساتید» اشتباه و درست آنها «پیشنه‌ادها» و «استادان» است.

بهتر است همواره حرف اضافه «به» از کلمه بعدی خود جدا نوشته شود، مگر آن که این حرف جزء یک فعل یا صفت یا قید باشد؛ مانند: «بکار بستن»، «بجا» و «بندرت».

در مورد کلمات حاوی همزه قواعدی وجود دارد که پرداختن به آنها در این مقاله نمیگنجد، اما برای نمونه به املاي کلمات «مسأله»، «منشأ» و «رئیس» دقت کنید. همچنین، همزه در انتهای کلماتی که به الف ختم میشوند، نوشته نمیشود و در صورت اضافه شدن به کلمه بعدی، از «ی» استفاده میشود: «اجرا شده»، و «اجرای برنامه».

۳-۶ شکل‌ها و جدولها

شکل‌ها و جدولها باید دارای عنوان باشند. عنوان شکل‌ها در زیر شکل و عنوان جدولها در بالای جدول قرار می‌گیرند. در صورتی که از شکل‌ها یا جدولهای سایر منابع استفاده می‌کنید، باید حتماً شماره آن مرجع را در عنوان شکل یا جدول ذکر کنید.

در هنگام ارجاع به شکل یا جدول از شماره آن استفاده کنید و از بکار بردن عباراتی همچون «شکل زیر» پرهیز کنید. تمام جدولها و شکل‌ها باید در متن مورد ارجاع قرار گیرند.

۴-۶ فرمول‌ها و عبارات ریاضی

برای هر فرمول باید يك شماره در نظر گرفته شود. این شماره را در داخل يك جفت هلالین و بصورت راست‌چین قرار دهید. در \LaTeX شماره گذاری به صورت خودکار انجام می‌پذیرد. تمام متغیرها، پارامترها، و نمادهای يك عبارت ریاضی باید توضیح داده شوند. اگر قبل از نوشتن فرمول این کار انجام نشده است، باید بلافاصله پس از فرمول این توضیحات بیان شوند. مثال زیر را در نظر بگیرید.

اگر ρ بیانگر چگالی تخمینی باشد، خواهیم داشت:

$$\rho = \frac{m}{A}, \quad (1)$$

که در آن m جرم تخمینی و A سطح آن است.

اگر در مقاله شما نمادهای ریاضی متنوعی مورد استفاده قرار می‌گیرد، حتماً سعی کنید جدول نمادها در متن خود داشته باشید. مثل جدول ۱.

جدول ۱: نمادهای مورد استفاده در مقاله

نماد	توضیحات
N	تعداد کل گره‌های شبکه
\mathbb{R}^{++}	مجموعه اعداد حقیقی مثبت بزرگتر از صفر
ρ_t	چگالی شبکه در لحظه t
$\Pr[A]$	احتمال رخداد رویداد A

۷ نکاتی در مورد نوشتن مقاله با \LaTeX

نویسندگانی که با محیط \LaTeX و نحوه کار با آن آشنایی ندارند می‌توانند به سایت www.parsilatex.com مراجعه کنند. مطالب مفید بسیاری در این زمینه در سایت مذکور قرار داده شده است. همچنین تمامی سوالات و اشکالات مرتبط با این استایل و در حالت کلی مرتبط با \LaTeX را می‌توانید در تالار پرسش و پاسخ همین سایت مطرح نمایید.

برای کار با این استایل در ابتدا چندین گام را می‌بایست بردارید:

۱. نصب یک کامپایلر به مانند TeXLive. دقت کنید این استایل را می‌توانید توسط TeXLive 2014 و یا TeXLive 2015 بروز شده استفاده نمایید. برای دانلود TeXLive می‌توانید به سایت TeXLive مراجعه کنید. در ضمن قابلیت خرید اینترنتی و ارسال پستی این مجموعه از طریق سایت parsilatex.com نیز فراهم شده است، بدین منظور به آدرس زیر مراجعه کنید.

بخش خرید سایت پارسی‌لاتک

۲. نصب یک ویرایشگر مناسب برای زبان فارسی به مانند TeXStudio. نصب فونت‌های IRMitra و Times New Romans.

۳. کامپایل فایل نمونه ارایه شده. نویسندگان در صورتی سه گام قبلی را با موفقیت پشت‌سر گذاشتند که بتوانند فایل نمونه قرار داده شده را با موفقیت و به صورت Normally کامپایل کنند.

لازم به ذکر است که تمامی عبارت‌ها و کلمات انگلیسی در فایل \LaTeX باید درون دستور `\Ir` قرار گیرند. مثلاً: Support vector و Machine Learning. دقت کنید که تمامی کلمات و عبارت‌های اختصاری می‌بایست در اولین فراخوانی به صورت باز شده ارایه شود. به عنوان مثال در اولین مکانی که شما از کلمه اختصاری SVM استفاده می‌کنید باید آن را به صورت SVM (support vec-tor machine) بنویسید. به صورت معمول تمامی حروف حالت باز شده باید با حرف کوچک نوشته شود.

ترجیحاً توصیه می‌شود که نویسندگان از قراردادن پاورقی در مقاله پرهیز کنند، اما در صورت نیاز شما می‌توانید با دستور footnote پاورقی فارسی و با دستور LTRfootnote پاورقی انگلیسی قرار دهید. مثل: در این روش^۲ و یا حسگر^۳. دقت کنید که همه پاورقی‌ها در انتهای متن در بخشی به نام پانویس چاپ خواهند شد و نه در همان صفحه.

در صورتی که می‌خواهید بر روی یک کلمه و یا عبارت در یک متن تأکید کنید، لطفاً از دستور `\emph` استفاده کنید، مثل: کار اصلی ما در این مقاله ارایه یک روش داده کاوی است. تقریباً تمامی بسته‌های مورد نیاز برای نوشتن یک مقاله در استایل

\LaTeX تهیه شده قرار داده شده است. اما در صورت نیاز به یک بسته معین لطفاً استایل قرار داده شده را تغییر دهید و قبل از فراخوانی بسته‌های `hyperref` و بسته `XqPersian` بسته خود را وارد کنید. در هنگام آپلود مقاله نیز کل فایل‌های \LaTeX باید به صورت zip شده در سایت مورد نظر قرار داده شود. محتویات فایل با پسوند zip باید شامل خروجی pdf، تمامی تصاویر، فایل اصلی tex، فایل با پسوند bib و هر فایل دیگر مورد نیاز باشد.

برای نوشتن فرمول یک خطی در \LaTeX به سادگی می‌توانید از محیط `equation` استفاده کنید. مثل:

$$A = B^2 + \frac{\gamma}{4}. \quad (2)$$

روابط می‌بایست در صورت نیاز حتماً شماره‌گذاری شوند. برای عدم شماره‌گذاری می‌توانید از محیط `equation*` بهره بگیرید. مثل:

$$A = \frac{\gamma}{\zeta}, \quad \gamma \in \mathbb{R}^{++}.$$

برای روابط چندخطی از محیط `align` استفاده کنید. مثل:

$$\begin{aligned} A &= B^c + \alpha \\ R &= A + T. \end{aligned} \quad (3)$$

و یا به عنوان مثال دیگر:

$$\begin{aligned} H(\lambda_g | \lambda_g + \lambda_d) &= \frac{1}{N} \sum_{i=1}^N \log_2(N - i + 1) \\ &+ \frac{1}{N} \sum_{i=1}^N \frac{\sum_{j=i}^N \log_2(\Upsilon_N - \Upsilon_{N-j})}{N - i + 1}, \\ &= \sum_{i=1}^N \frac{\sum_{j=i}^N \log_2(\Upsilon_N - \Upsilon_{N-j})}{N(N - i + 1)}. \end{aligned} \quad (4)$$

برای ارجاع به روابط و فرمول‌ها نیاز نیست بنویسید مثلاً فرمول شماره ... فقط کافی است که برچسب فرمول را با دستور `eqref` فراخوانی کنید. در این صورت نیازی به قرار دادن پرانتز نیست و \LaTeX به صورت خودکار پرانتزها را می‌گذارد. مثلاً با قرار دادن (۳) در (۲) خواهیم داشت ...

سعی کنید مطالب خود را به صورت منظم در قالب تعدادی تعریف، قضیه، لم و گزاره بیاورید. تمامی این محیط‌های در استایل آماده شده تعریف شده است و براحتی قابل استفاده است.

تعریف ۱ (حالت پایدار). حالتی را پایدار گوئیم که در آن تغییر آمارگان پارامترهای صف برابر صفر باشد. اگر ϖ را یک پارامتر صف در نظر بگیریم، خواهیم داشت.

$$\frac{d\varpi(t)}{dt} = 0 \quad (5)$$

قضیه ۱ (پایستگی جریان). در صورتی که سامانه مورد بررسی (رگودیک و در پایدار باشد، آن گاه نرخ ورود بسته به سامانه همواره برابر با نرخ خروج آن خواهد بود.

قضیه ۲. قرار دادن نام برای قضیه به مانند قضیه قبل که نامش پایستگی جریان بود اختیاری است.

اثبات. سعی کنید اگر قضیه برای مرجع دیگری است اثبات آن را در مقاله بیاورید و فقط به مرجع مذکور ارجاع دهید. اگر اثبات در روند مقاله اهمیت دارد آن را درون

متن مقاله بیاورید. اما اگر در روند کلی تأثیری ندارد اثبات را به قسمت پیوست‌ها ببرید. □

نتیجه ۱. در یک صف نرخ ورود با خروج برابر است.

نتیجه ۲. در یک سامانه نرخ ورود برابر است با تعداد بسته‌ها

لم ۱. اگر N_t نشان‌دهنده تعداد بسته رسیده تا زمان t به

$$P(t_i < t) = 1 - P(N(t) < i) \implies P(t_i > t) = P(N(t) < i) \quad (6)$$

اثبات لم. برای اثبات به مرجع برای اثبات قضایا از محیط `proof` و برای اثبات لم‌ها از محیط `lemmaproof` استفاده کنید. ■

اصل ۱ (عدم قطعیت). برطبق اصل عدم قطعیت هر ذره

توصیه می‌شود که نویسندگان برای نمادهای ریاضیاتی سعی کنند از نمادهای ساده و استاندارد استفاده کنند. به عنوان مثال مجموعه اعداد حقیقی را بهتر است به جای R با \mathbb{R} نشان داد. تمامی عملگرهای ریاضیاتی به مانند عملگر امیدریاضی، آنتروپی، احتمال رخداد یک رویداد باید به صورت غیرایتالیک نوشته شود، مثل: $H[\dots]$, $\Pr[\dots]$, $E[\dots]$ و ...

نویسندگان می‌بایست حتماً و حتماً نمادها و روابط ریاضی موجود در متن را در حالت `math mode` بنویسند و نه در حالت متنی. به عنوان مثال به جای $A+B$ که به صورت متنی است، بهتر است بنویسید $A + B$.

۸ نتیجه گیری

در این بخش نویسندگان باید به صورت خلاصه کل روندی که در مقاله پیموده شده است را توضیح دهند. در ضمن نویسندگان می‌توانند در این بخش ایده‌های جدید برای توسعه هرچه بیشتر و بهتر مقاله خود را مطرح کنند.

سپاس‌گزاری

بخش سپاس‌گزاری در صورت نیاز بصورت کوتاه و در یک بند آماده شود. بخش سپاس‌گزاری دارای شماره نیست. در این قسمت نویسندگان می‌توانند از افراد و نهادهای پشتیبان و یاریگر تشکر و قدردانی کنند. از همین مجال استفاده می‌شود و از همه دوستانی که در روند برگزاری کنفرانس و تهیه این استایل یاریگر ما بودند، تشکر و قدردانی به عمل می‌آید، به خصوص آقایان وفا خلیلی نویسنده بسته `XqPersian`، محمد امین طوسی، امیر حسین رضایی تبار، وحید دامن‌افشان، فرشاد ترابی و دیگر دوستان در سایت پارسی‌لاتک.

پیوست‌ها

بخش پیوست‌ها یک بخش اختیاری است و شماره‌گذاری نمی‌شود. موضوع‌های مرتبط با مقاله که در یکی از گروه‌های زیر قرار گیرند، می‌توانند در بخش ضمائم آورده شوند.

- اثبات ریاضی فرمول‌ها یا الگوریتم‌ها.
- داده‌ها و اطلاعات مربوط به مطالعه موردی.
- نتایج کار دیگر محققان و داده‌های مربوط به مقایسه آن‌ها.
- سایر موضوع‌های مرتبط که جزء بخش‌های اصلی مقاله نباشند.

مراجع

- [1] [Official HTML 4.0 Specification](#). W3C Recommendation, revised on 24-Apr-1998.
- [2] Wikipedia: [Hypertext Markup Language](#).
- [3] [Cons of HTML](#)
- [4] Medium Article, [Pros of HTML](#), olivia sand, Jan 2018.
- [5] Choose your app platform, [Article](#), Microsoft, 2019.
- [6] [Android Layouts](#), Android Developers Document, Google.
- [7] [Article](#), 9 ways to avoid memory leaks in Android, Anitaa Murthy, May 2018.
- [8] [ProAndroidDev Article](#), Everything you need to know about Memory Leaks in Android, Ali Asadi, Jun 2019.
- [9] [Android Jetpack](#), Android Developers Document, Google.
- [10] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2006.
- [11] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [12] R. T. Durrett. *Essentials of Stochastic Processes*. Springer Texts in Statistics, 2 ed. , 2012.
- [۱۳] مهاجری، جواد، باباحیدریان، پریسا، و دیانت، ابوالفضل، "طرح جدید مدیریت کلید آستانه ای در شبکه های سیار اقتضایی،" در شانزدهمین کنفرانس ملی سالانه انجمن کامپیوتر ایران، ۱۳۸۹.
- [14] J. Yang and J. Leskovec, "Modeling Information Diffusion in Implicit Networks," *IEEE International Conference on Data Mining*, pp.599–608, Dec. 2010.
- [15] J. Beran, R. Sherman, M. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on Communications*, vol.43, no.2, pp.1566–1579, 1995.

پانویس ها

^۱ Anchor Node

^۲ در واقع منظور ما ...

^۳ Sensor