3rd INTERNATIONAL CONFERENCE ON RESEARCH IN SCIENCE AND TECHNOLOGY
BERLIN . GERMANY 9 July, 2016
RSTCONF
سومین کنفرانس بین المللی پژوهش در علوم و تکنولوژی
برلین - آلمان ۱۹ تیرماه ۱۳۹۵

بررسی روش‌های راهنمایی و مدیریت کد در محیط برنامه‌نویسی

ایوب چادرلاجوردی
دانشجوی کارشناسی ارشد دانشگاه آزاد اسلامی واحد ملایر
ayooblajavardi@gmail.com

چکیده

در زبان‌های برنامه نویسی شی‌گرا به صورت‌های مختلف می‌توان کد را ارائه داد و آن کد را اجرا ونتیجه نهایی مشاهده نمود و در نهایت نتایج را بررسی کرده و اصلاحات لازم و مورد نیاز را بر روی آن انجام داد کدهای ارائه شده توسط برنامه‌نویسان به روشهای مختلفی ارائه می‌شوند این کدها گاهی کاملا به صورت اتوماتیک و خودکار تولید می‌شوند البته باید قسمت طراحی با یک زبان فرمال انجام گیرد. در زبانهای برنامه نویسی فعلی کدها به صورت‌های مختلفی ارائه می‌شوند و هر کدام از آنها مزایایی دارند و البته می‌توان بخش‌های مختلف هر قسمت را با روش‌های دیگر ادغام نمائیم و در نهایت باعث بهبود سرعت در ایجاد کد شویم. اما باید توجه داشته باشیم که علاوه بر سرعت باید میزان کارایی کد را نیز در نظر داشته باشیم کدها ممکن است از نظر سرعت و میزان منابع مورد نیاز نیز مورد بررسی قرار گیرند

واژگان کلیدی: تولید کد – کدهای اتوماتیک – کدهای تولید شده

مقدمه

کدهای نوشته شده در محیط‌های برنامه‌نویسی گاهی به صورت اتوماتیک ایجاد شده و برنامه نویسان کدهایی را به آن اضافه می‌کنند که این کد با توجه به نیاز مسئله اضافه می‌شوند اما این کدها در چه مرحله‌ای تولید می‌شوند و چگونه باید توسط کامپایلر کامپایل شوند و در چه زمانی ایجاد می‌شوند با توجه به زبان و محیط برنامه نویسی متفاوت است و یا حتی اینکه سخت‌افزارهای مورد نیاز برای این کدها باید چه باشند و چگونه از این سخت افزارها استفاده می‌کنند در این مقاله سعی می-کنیم انواع روش‌های تولید کد اتوماتیک را بررسی کنیم و تاثیر بهبود کدنویسی و میزان سرعت کد نویسی را بررسی کنیم. محیط‌های برنامه‌نویسی فعلی که برنامه‌نویسان در آن بیشتر احساس راحتی دارند این گونه ویژگی‌ها را بیشتر پشتیبانی می کنند. با این‌گونه کد نویسی، برنامه نویسان زمان کمتری را برای نوشتن کد صرف کرده و بیشتر دقت آنها برای قسمت طراحی و نحوه‌ی ایجاد ارتباط بخش‌های مختلف صرف می‌شود که این امر باعث بهبود در تولید کانکتورها نیز می‌شود.

برنامه نویسی چیست

برنامه نویسی رایانه‌ای در دید افراد غیر متخصص تمام امور تولید یا بهبود یک نرم‌افزار را برنامه نویسی می‌گویند اما در واقع تبدیل خواسته ها و نیازمندی‌ها به کد را برنامه نوسی می‌گویند که این برنامه نویسی با زبان‌های مختلف و در سطوح مختلف انجام می‌شود و بعد از ایجاد کد تست و نگهداری نیز انجام می‌شود. البته این موارد از مرحله کد نویسی مجزا می‌باشند. در کل برنامه نویسی عبارت است از نوشتن یک سری دستورات که کامپیوتر آنها را فهمیده و آن را اجرا می‌کند. (fa.Wikipedia.org)

کد نویسی

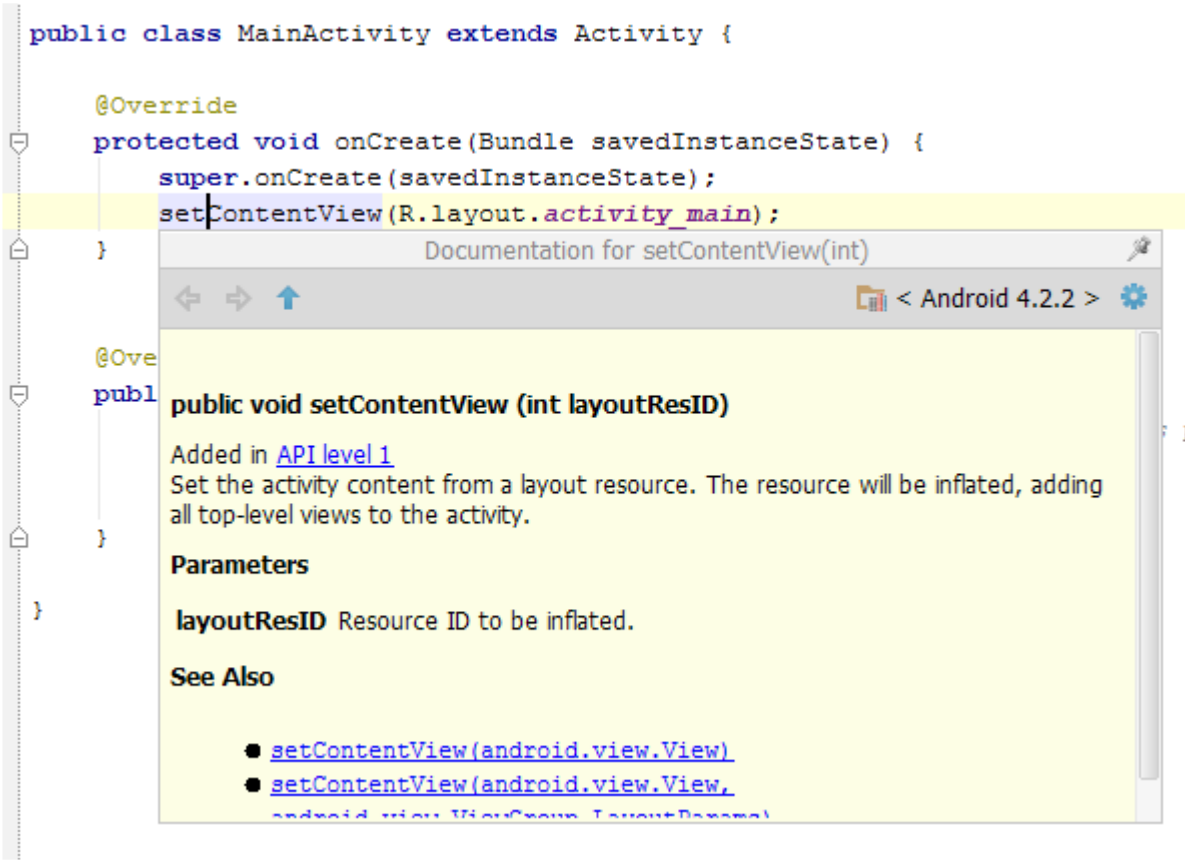
برنامه نویسان تا چند سال اخیر تمامی کدها را به صورت خط به خط خود تولید و از آن استفاده می‌کردند و کمی بعد برای کدهای خود توضیحاتی را اضافه نمودند تا بتوانند در مدتها بعد برای بهبود کارایی کد نوشته شده اقدام کنند یا اینکه بتوانند اصلاحاتی را انجام بدهند این نوشتن کد خود خیلی زمان‌بر بود و گاهی حتی به دلیل وجود اشتباهات در تایپ روزها برای رفع خطای خود اقدام می‌کردند. یا اینکه بایست کدها را کامل تایپ نمایند. و بعد از آن باید شکل کامل و صحیح هر کد و تابع را در خاطر داشته باشند که این امر در پروژه‌های بزرگ کار بسیار سخت و نا ممکن به نظر می‌اید. حال اگر شخصی این کار را هم انجام دهد اما بعد از چند مدت شکل صحیح کد را فراموش می‌کرد . پس اینگونه کد نویسی در دراز مدت بسیار خسته کننده بود و در زمان پشتیبانی بسیار کار کند پیش می‌رود و از مشکلات برنامه نویسان نوشتن کامل کدها بود که در پروژه‌های بزرگ عذاب آور بود.

۱- تولید کد

در چند دهه اخیر کدها دیگر نیازی به تایپ کامل ندارند بلکه برنامه‌نویس بخشی از آن را تایپ کرده و سیستم خود به صورت خودکار یک لیست از کدهای پیشنهادی را به برنامه نویس ارائه می‌دهد و حتی در زمان ایجاد پروژه قسمتهای اتصال و مهم برنامه را که حالت کلی دارد برای برنامه‌نویس طراحی و ایجاد می‌کند . حتی محیط‌های برنامه‌نویسی مختلف این کدها را کلاس‌بندی و طبقه بندی می کنند و یا اینکه کد را در زمان تولید و تایپ کامپایل می‌کنند و خروجی همان خط کد را به صورت آنی ملاحظه می‌کنند. اما کدهای ایجاد شده توسط محایط برنامه نویسی بسیار متناوب می‌باشد ولی باید توجه داشته باشیم که در یک محیط برنامه نویسی پیشرفته و نوین تمامی این موارد ممکن است وجود داشته باشد

۱-۱- نمایش سریع مستندات با حرکت موس

زمانی که موس را بر روی یک تابع حرکت دهیم و کمتر از ۵۰۰ میلی ثانیه متوقف کنیم سریعا یک توضیح چند جمله‌ای درباره‌ی آن ظاهر می‌شود و راهنمایی مفید برای برنامه نویس می‌باشد این گونه مستندات در زمان اطلاحات بسیار مفید بوده و سرعت انجام عمل را بالا برده و به هیچ عنوان توسط پردازنده در هنگام اجرا کامپایل نشده و برنامه نویس خود هیچ دخلی در تولید آن نداشته است و سیستم خود با تشخیص خود آن را ایجاد کرده و نمایش می‌دهد. حال اگر موس را تکان دهیم دوباره توضیحات محو شده و دیگر قابل رویت نخواهند بود اما اگر برنامه نویس بخواهد توضیحات بیشتری را در مورد آن کد بداند موس را بر روی توضیحات ظاهر شده قرار می‌دهد و توضیحات بیشتری را راجع به آن کد در راهنما مطالعه و مشاهده می-نماید.(شکل ۱)(Belen Cruz Zapata – 2103)



شکل ۱ – قرار گیری موس روی کد و نمایش توضیحات مورد نیاز برای برنامه نویس

۲- کلیدهای هوشمند

زمانی که ما کدهایی را تایپ می‌کنیم نیاز به تعیین محدوده برای انجام و تاثیر گذاری هر قطعه کد می‌باشیم که برای اینکار از براکت یا پرانتز یا هر چیز دیگری استفاده می‌کنیم که معمولا توسط برنامه نویسان این امر به دلیل ناخودآگاهی و به دلیل خستگی یا هر عامل دیگری فراموش می‌شود و تا زمان کامپایل کد متوجه آن نمی‌شوند و ممکن است این عمل زمانی رخ دهد که هزاران منطقه در برنامه ایجاد شده باشد حال این منطقه ایجاد شده به چه وسعت است نیاز به بررسی یک دور کامل همه برنامه دارد که برای برنامه نویسان بسیار زمان بر و طاقت فرساست. در بعضی زبانهای برنامه نویسی و در بعضی محیطها در

لحظه ایجاد یک محدوده سیستم به صورت خودکار محدوده را شناسایی کرده و آن را اعلام می‌دارد و برنامه نویس زمان خود را فقط صرف نوشتن کد می‌کند و ایجاد و کنترل محدوده توسط سیستم انجام می‌شود

۳- رنگ‌ها و قلم‌ها

در محیط‌های برنامه نویسی امروزی کدها هرکدام با رنگ خاصی مشخص می‌شوند و هر بخش با یک اندازه خط خاصی در صفحه نمایش داده می‌شوند تا برنامه نویس بتواند بهتر آن را بررسی کند و اینکه سیستم این عمل تفکیک رنگ را انجام می‌دهد و اندازه‌ی قلم هر بخش و هر سر عنوان را انجام می‌دهد پس با این عمل سیستم از تمامی تمایزات برخوردار می‌شود و برنامه نویس در هر زمان که بخواهد با یک صفحه رنگ بندی شده و کدهای با اندازه‌های مختلف روبه‌رو خواهد شد که به صورت استاندارد این وضعیت مشخص می‌شود و در هر زمان که برای بررسی کد و خطا و یا ویرایش مراجعه کند سریعتر کارها را انجام می‌دهد

۴- نمایش درختی

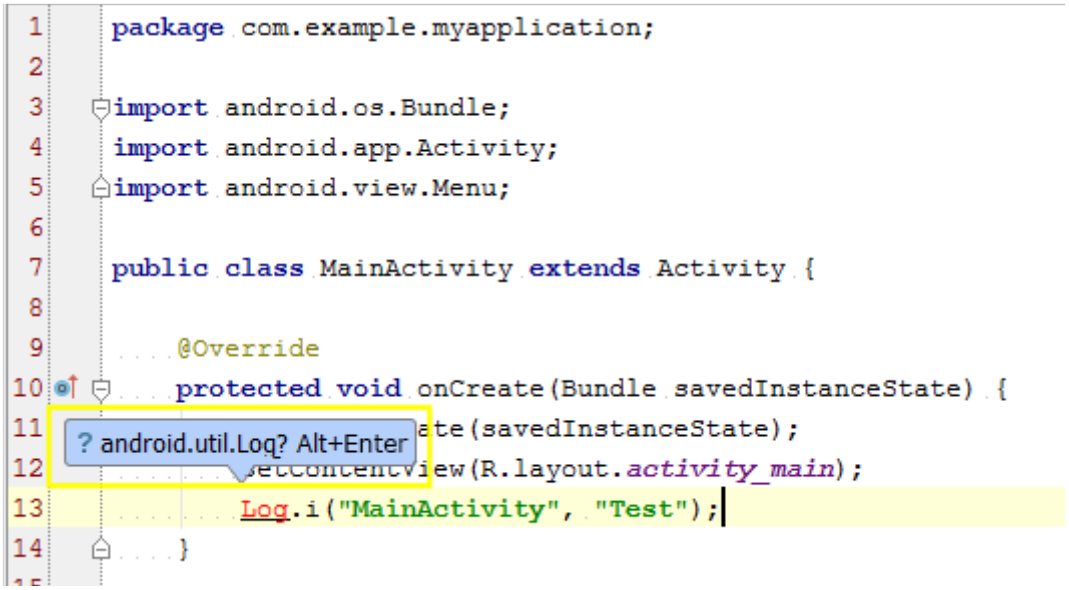
در بعضی از محیط‌های برنامه نویسی کدها به صورت یک منوی باز شونده در هر بخش قابل رویت و مخفی شدن هستند. زمانی که کدهای یک تابع به طور کامل نوشته و بازبینی می‌شوند دیگر نیازی به وجود در محل ندارند و میتوانیم آنها را از دید خارج کنیم که این عمل code folding نامیده می‌شود. کد که در همان محل مخفی میشود به صورت یک علامت مثبت نمایش داده می‌شود و برنامه نویس و کدنویس با تسلط بیشتری بر صفحه می‌تواند ادامه عملیات را داشته باشد.(شکل ۲) (https://en.wikipedia.org)

```
1   package com.example.myapplication;
2
3   import ...
6
7   public class MainActivity extends Activity {
8
9       @Override
10      protected void onCreate(Bundle savedInstanceState) {
11          super.onCreate(savedInstanceState);
12          setContentView(R.layout.activity_main);
13      }
14
```

شکل ۲- نمایش کدها به شکل folding

۵- اضافه کردن کد

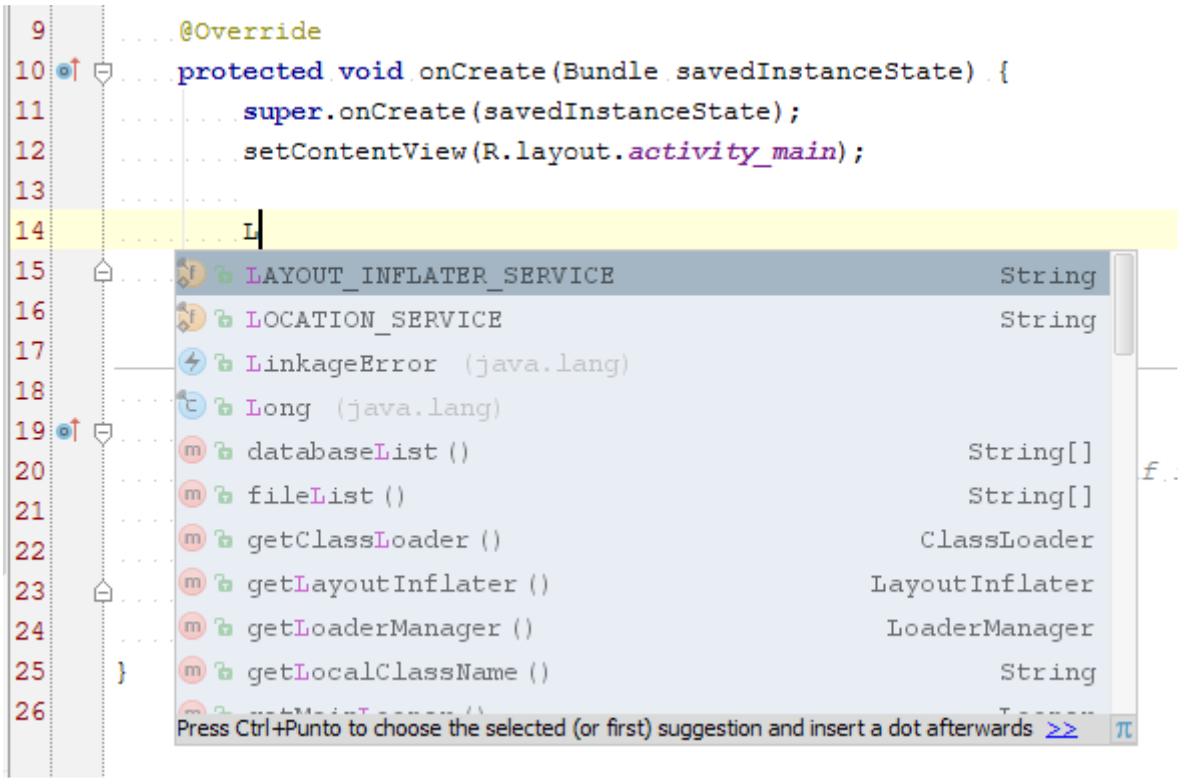
هنگامی که کدهای یک بخش را کپی گرفته و به بخش دیگری اضافه می‌کنیم ممکن است بخش‌هایی که به عنوان پیش نیاز است را هنوز انجام نداده باشیم حال برنامه نویس باید تمامی کد را بررسی و منابع مورد نیاز برای تکه کد را نیز با خود حمل کند و در برنامه مقصد قرار دهد تا بتواند به طور صحیح و بدون مشکل از آن استفاده کند اما در محیط‌های برنامه نویسی جدید این مشکل حل شده و در زمان قرار دادن کد در محل جدید کامپایلر نیازهای اساسی کد مهمان را بررسی کرده و در برنامه منبع فراخوانی کرده و به عنوان یک کلاس و یا یک تابع در برنامه خود قرار می‌دهد و به این‌صورت تبادل کد بین چندین برنامه به صورت اضافه شدن نیازهای بنیادین بالا رفته و تمام زمان برنامه‌نویس صرف نوشتن کد جدید می‌شود. و این نوع کد نویسی مثال بارزی از استفاده مجدد از کدها می‌باشد. حتی در بعضی محیط‌های برنامه نویسی چند خط توضیحات جهت راهنمایی برای مراجعات بعدی نیز افزوده می‌شود.(شکل ۳)(https://en.wikipedia.org/wiki/Autocomplete)



شکل۳- تعیین کدهای انتقالی از یک بخش دیگر

۶- Code completion

در این نوع کدنویسی سرعت کدنویسی بسیار بالا رفته و علاوه بر آن میزان خطا در تایپ نیز بسیار کاهش پیدا می‌کند در این روش برنامه‌نویس با تایپ یک کاراکتر یک لیست از دستورات و اشیای مورد نیاز که با آن کاراکتر آغاز می‌شوند را در اختیار می‌گیرد و به آسانی و سادگی می‌تواند موارد مورد نظر خود را انتخاب کند و ادامه کدنویسی و تایپ برای آن خط کد را متوقف کند و در سطح های پیشرفته‌تر این نوع کد نویسی حتی اگر زمانی دو شی همنام با مشخصه‌های متفاوت داشته باشیم شی مورد نظر را نیز تشخیص می‌دهد که این امر بیشتر در شناسایی آرگومانهای تابع کاربرد دارد.(شکل ۴)
(https://en.wikipedia.org/wiki/Intelligent_code_completion)



شکل ۴ – نحوه استفاده از تکمیل کننده کد

۷- نتایج

با بررسی کدها به صورت اتوماتیک میزان خطا بسیار کاهش پیدا کرده و دیگر نیاز به اجرای آزمایشی برنامه نیست چرا که اینکار در برخی موارد ممکن است صرف هزینه زیادی داشته باشد و با تولید کد به صورت خودکار دیگر نیازی نیست که برنامه‌نویس زمان زیادی را جهت تایپ برنامه و کد صرف کند بلکه بیشتر زمان برای نحوه‌ی پیاده سازی صرف می‌شود و این یک فرصت برای ایجاد یک کانکتور قوی در بین اجزای مختلف برنامه می‌باشد که خود ممکن است یک الگوی خاص برای حل مسئله ارائه دهد و در نتیجه سرعت و کیفیت اجرای برنامه نیز بهبود یابد. با داشتن محیط برنامه نویسی خودکار در بررسی کد در زمان‌های بروزرسانی موفق‌تر خواهیم بود و زمان کمتری را صرف بررسی کدها می‌کنیم.

منابع

http://fa.wikipedia.org/wiki/برنامه نویسی
Belen cruz zapta – android studio application development . october 2013
https://en.wikipedia.org/wiki/Code_folding
https://en.wikipedia.org/wiki/Autocomplete
https://en.wikipedia.org/wiki/Intelligent_code_completion