



آموزش BlueJ

Persian Translator: Mehdi shahdoost

Persian Support by: JEIC SOFT (Jafzar Engineering and Investigation Company)

**Michael Kölling
Mærsk Institute
University of Southern Denmark**

فهرست مطالب

صفحه

فصل ۱ – مقدمه

۵ ۱-۱) درباره BlueJ.....
۵ ۲-۱) مطالب پوشش داده شده توسط کتاب و خوانندگان.....
۵ ۳-۱) حق تالیف ، ليسانس و توزيع مجدد.....
۵ ۴-۱) بازتاب و واکنش شما.....

فصل ۲ – نصب کردن

۶ ۱-۲) نصب در ویندوز.....
۷ ۲-۲) نصب در مکینتاش.....
۷ ۳-۲) نصب در لینوکس / یونیکس و سیستم های دیگر.....
۷ ۴-۲) مشکلات نصب کردن.....

فصل ۳ – شروع کردن (ویرایش کردن- کامپایل کردن – اجرایی کردن)

۸ ۱-۳) شروع BlueJ.....
۹ ۲-۳) باز کردن پروژه.....
۹ ۳-۳) ایجاد اشیا.....
۱۱ ۴-۳) اجرایی کردن.....
۱۳ ۵-۳) ویرایش کلاس ها.....
۱۴ ۶-۳) کامپایل کردن.....
۱۵ ۷-۳) راهنمایی با استفاده از خطاهای کامپایلر.....

فصل ۴ – انجام کارهای بیشتر

۱۶ ۱-۴) بررسی کردن و بازبینی.....
۱۸ ۲-۴) ارسال اشیا به عنوان پارامترها.....

۲۰	فصل ۵ – ایجاد یک پروژه جدید
۲۰ (۱-۵) ایجاد پوشه پروژه
۲۰ (۲-۵) ایجاد کلاس ها
۲۰ (۳-۵) ایجاد وابسته ها
۲۱ (۴-۵) جابجا کردن عناصر (elements)

۲۲	فصل ۶ – استفاده از محیط کدنویسی
۲۲ (۱-۶) نمایش code pad
۲۲ (۲-۶) ارزیابی عبارات ساده
۲۳ (۳-۶) دریافت اشیا
۲۴ (۴-۶) بررسی اشیا
۲۴ (۵-۶) اجرایی کردن دستورات
۲۴ (۶-۶) دستورات چند خطی و توالی دستورات
۲۴ (۷-۶) کار با متغیرها
۲۵ (۸-۶) تاریخچه دستورات

۲۶	فصل ۷ – برطرف کردن خطاها
۲۶ (۱-۷) تنظیم کردن Breakpoint ها
۲۸ (۲-۷) عبور مرحله ای از میان کدها
۲۸ (۳-۷) بررسی متغیرها
۲۹ (۴-۷) توقف کردن و تمام کردن

۳۰ **فصل ۸ – ایجاد برنامه های کاربردی**

۳۲	فصل ۹ - ایجاد اپلت ها
۳۲ (۱-۹) اجرا کردن اپلت
۳۳ (۲-۹) ایجاد کردن اپلت

۳۳ تست کردن اپلت (۳-۹)

فصل ۱۰ - اعمال دیگر

۳۴

۳۴ باز کردن پکیج های غیر BlueJ در BlueJ (۱-۱۰)

۳۴ اضافه کردن کلاس های موجود به پروژه تان (۲-۱۰)

۳۴ فراخوانی متدهای استاتیک و main (۳-۱۰)

۳۵ تولید اسناد (Documentation) (۴-۱۰)

۳۵ کار با کتابخانه ها (۵-۱۰)

۳۵ ایجاد اشیا از کلاس های کتابخانه ای (۶-۱۰)

فصل ۱۱ - فقط خلاصه ها

۳۶

مقدمه

۱-۱) درباره BlueJ

این آموزش مقدمه ای برای استفاده از محیط برنامه نویسی BlueJ میباشد. BlueJ محیط توسعه جاوا میباشد که به طور ویژه برای آموزش جاوا در سطوح مقدماتی طراحی شده است. این نرم افزار توسط تیم طراحی و پیاده ساز در دانشگاه Deakin شهر ملبورن استرالیا و همچنین دانشگاه کنت (Kent) شهر کانتربری انگلستان ایجاد شده است. برای بدست آوردن اطلاعات بیشتر به وب سایت این نرم افزار به آدرس زیر مراجعه نمایید :

[Http://www.bluej.org](http://www.bluej.org)

۲-۱) مطالب پوشش داده شده و خوانندگان کتاب

این آموزش برای تعلیم دادن جاوا نمی باشد و در اصل محیطی برای آموزش جاوا می باشد.

۳-۱) حق تالیف ، ليسانس و توزيع مجدد

نرم افزار BlueJ و همچنین این آموزش همانگونه که هستند (as-is) برای استفاده های شخصی و غیر تجاری بصورت رایگان ارائه می شوند. حق تالیف BlueJ در اختیار M. Kölling and J. Rosenberg قرار دارد.

۴-۱) بازتاب و واکنش شما

نظرات ، سوالات ، اصلاحات و انتقادات شما در خصوص نرم افزار و یا آموزش BlueJ باعث خوشحالی و تشویق ما خواهد شد. لطفاً میل کنید به مایکل کولینگ (Michael Kolling) به آدرس زیر :

mik@mip.sdu.dk

۵-۱) سخن مترجم

آموزشی که همانکون در اختیار دارید ترجمه متن انگلیسی آموزش BlueJ می باشد. که در سایت رسمی این نرم افزار موجود است. البته این متن ترجمه لغت به لغت آموزش انگلیسی نرم افزار نیست بلکه برداشتی آزاد از آن متن است که کوشش شده تمام متن اصلی را پوشش دهد. امیدوارم که این اثر که از طرف شرکت تحقیقاتی و مهندسی جی افزار پشتیبانی می شود مقبول شما برنامه نویسان و توسعه دهندگان ایرانی قرار بگیرد. در صورتی که شما دوستان عزیز کاستی در این اثر دیدید لطفاً برای برطرف کردن آن در نسخه های بعدی این آموزش آنرا با استفاده از ایمیل زیر به بنده اطلاع دهید.

Persianshadow2003@yahoo.com

فصل دوم - نصب کردن

BlueJ در سه فرمت توزیع شده است: یکی برای سیستم عامل ویندوز ، یکی برای سیستم عامل مکینتاش و یک نسخه هم برای دیگر سیستم ها . مراحل نصب این نرم افزار کاملا سر راست و مشخص می باشد.
پیش نیازها :

شما برای استفاده از BlueJ باید J2SE1.4 یا نسخه های بالاتر را نصب داشته باشید. ما توصیه می کنیم که از نسخه های غیر بتا (Beta) جاوا استفاده کنید. اگر شما جاوا (JDK) را ندارید. می توانید از طریق وب سایت شرکت سان آنرا بصورت رایگان دریافت کنید. :

<http://java.sun.com/j2se>

در سیستم عامل MacOS X آخرین نسخه جاوا بصورت پیش فرض نصب میباشد. و شما نیاز به نصب مجدد جاوا توسط خودتان ندارید. در ضمن اگر در صفحه دانلود شما با دو پیشنهاد JRE و یا SDK برخورد کردید. شما باید SDK را دانلود کنید. و JRE کافی نمی باشد.

۱-۲) نصب برای ویندوز

فایل توزیع شده برای سیستم عامل ویندوز با این نام می باشد : **bluejsetup-xxx.exe** . که کلمات xxx شماره نسخه BlueJ را مشخص می کند. به عنوان مثال فایل نصب کننده BlueJ نسخه 2.0.0 بدین صورت میباشد : **bluejsetup-200.exe** .

شما این فایل رو می توانید از طریق یک دیسک دریافت کرده و یا اینکه از وب سایت BlueJ دریافت کنید. با استفاده از این نصب کننده (installer) شما می توانید. پوشه ای را که قصد دارید. BlueJ در آن نصب شود مشخص کنید. همچنین این نصب کننده گزینه های برای نصب میانبر (shortcut) در منوی شروع و دسکتاپ به شما ارائه می دهد.

بعد از اینکه نصب تمام شد. شما می توانید برنامه **BlueJ.exe** را در پوشه ای که BlueJ نصب شده پیدا کنید.

اولین باری که شما برنامه BlueJ را اجرا کنید. BlueJ کامپیوتر شما را برای پیدا کردن جاوا (JDK) جستجو

می نماید. و اگر چند ورژن مناسب از جاوا (به عنوان مثال **JDK1.4** و **JDK1.5**) را روی سیستم شما پیدا کرد.

پنجره ای را باز می کند که شما می توانید یکی از آن نسخه های جاوا را انتخاب کنید.

در ضمن اگر BlueJ نتوانست. JDK را روی سیستم شما پیدا کند. مسیر آنرا از شما سوال خواهد کرد. (دلیل این امر

می تواند این باشد که JDK روی سیستم شما نصب شده باشد. ولی مقدار مورد نیاز از Registry ویندوز پاک شده باشد.

نصب کننده BlueJ همچنین برنامه **vmselect.exe** را نصب می کند. و شما با استفاده از این برنامه می توانید نسخه

جاوایی که BlueJ از آن استفاده می کند را تغییر دهید.

۲-۲) نصب برای مکینتاش

لطفاً توجه داشته باشید که BlueJ تنها در MacOS X اجرا می شود.

فایل های توزیع شده برای مکینتاش به شکل blueJ-xxx.zip نامیده می شوند، xxx ها شماره نسخه نرم افزار را مشخص می کنند. برای مثال نسخه blueJ 2.0.0 ، فایل های توزیع شده اش بصورت BlueJ-200.zip میباشد. شما این فایل را می توانید در یک دیسکت دریافت کرده و یا از وب سایت BlueJ به آدرس زیر دانلود نمایید:

<http://www.bluej.org>

سیستم عامل MacOS بعد از دانلود فایل بصورت خودکار فایل را از حالت زیپ شده خارج می کند. و اگر این کار صورت نگرفت، بر روی فایل دوبار کلیک کنید تا فایل از حالت زیپ خارج شود. بعد از این که فایل از حالت فشرده خارج شد، شما پوشه ای با نام BlueJ-xxx خواهید داشت. این پوشه را به پوشه برنامه های کاربردی تان منتقل نمایید (یا هر پوشه ای که تمایل دارید در آن نگه دارید). شما نیازی به نصب ندارید.

۳-۲) نصب کردن برای لینوکس ، یونیکس و سیستم های دیگر

فایل های عمومی توزیع شده برای اینکار یک فایل Jar اجرایی می باشد. این فایل BlueJ-xxx.jar خوانده می شود. که در آن xxx شماره نسخه نرم افزار میباشد. به عنوان مثال فایل های توزیع شده BlueJ 2.0.0 با نام BlueJ-200.jar مشخص می باشد. شما می توانید این فایل را در یک دیسکت و یا از طریق وب سایت BlueJ به آدرس زیر دریافت نمایید:

<http://www.bluej.org>

برنامه نصب کننده را با اجرای دستور زیر فراخوانی کنید. توجه: ما برای این مثال از فایل توزیع شده BlueJ-200.jar استفاده کرده ایم - شما نیاز دارید تا از نام فایلی که در یافت کرده اید استفاده کنید.

```
< j2se-path>/bin/java -jar bluej-200.jar
```

< j2se-path> مسیر پوشه ای میباشد که در آن J2se نصب میباشد. پنجره ای باز می شود که به شما اجازه می دهد تا محل نصب برنامه و همچنین نسخه جاوای که برای اجرایی BlueJ می خواهید از آن استفاده کنید تعیین نمایید. دکمه Install را فشار دهید. بعد از اتمام ، BlueJ بر روی سیستم شما نصب شده است.

۴-۲) اشکالات در نصب برنامه

اگر شما با مشکلی برخورد کردید. می توانید به بخش FAQ سایت BlueJ مراجعه کنید :

<http://www.bluej.org/help/faq.html>

و یا اینکه بخش " پرسش برای کمک " را مطالعه نمایید:

<http://www.bluej.org/help/ask-help.html>

فصل سوم - شروع کردن - ویرایش کردن - کامپایل کردن - اجرایی کردن

۱-۳) آغاز به کار با BlueJ

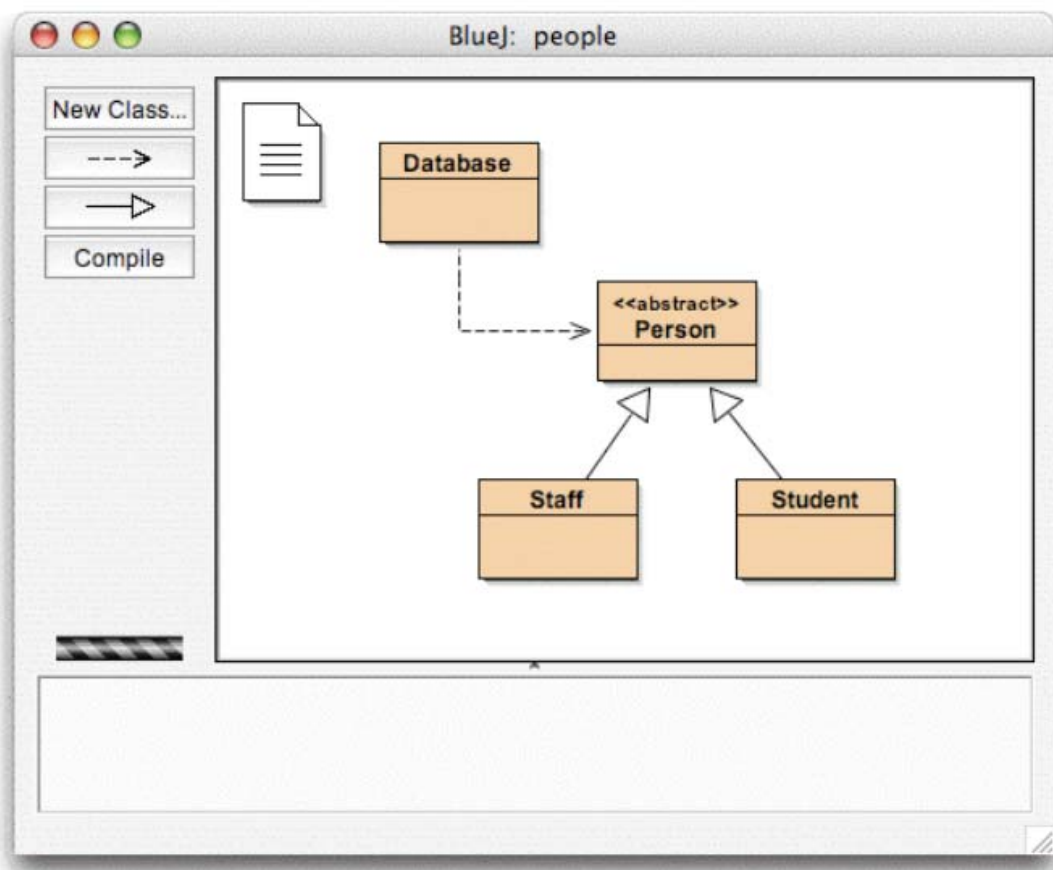
در سیستم عامل ویندوز و مکینتاش برنامه با نام BlueJ نصب شده . برای شروع شما باید این برنامه را اجرا کنید.
در سیستم های یونیکس برنامه نصب کننده اسکریپتی با نام BlueJ را نصب کرده که شما برای اجرای محیط گرافیکی روی این فایل دو بار کلیک کنید. در ضمن با استفاده از خط فرمان هم می توانید BlueJ را با آرگومان و یا بدون آرگومان فراخوانی کنید:

\$ BlueJ

و با آرگومان :

\$ BlueJ examples/people

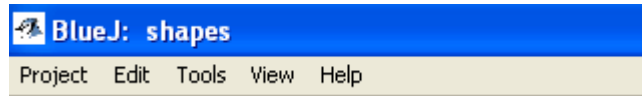
منظور از آرگومان هم پروژه ای است که قصد دارید با BlueJ باز شود هست.



(عکس ۱ : پنجره اصلی برنامه BlueJ)

۳-۲) باز کردن پروژه

به طور خلاصه برای باز کردن پروژه از منوی **Project** گزینه **Open** را انتخاب نمایید.



پروژه های **BlueJ** مانند پکیج های استاندارد جاوا از پوشه های که شامل فایل های پروژه هستند تشکیل شده است . بعد از اینکه برنامه **BlueJ** باز شد. با استفاده از منوی **Project** و گزینه **Open** پروژه مورد نظر خود را باز کنید. چند پروژه نمونه به همراه **BlueJ** توزیع شده است که شما این پروژه های نمونه را می توانید در پوشه **examples** پیدا نمایید.

برای این بخش آموزش شما پروژه **People** را که در پوشه **example** قرار دارد باز کنید. پوشه **example** در پوشه ای که **BlueJ** نصب شده قرار دارد. بعد از اینکه شما پروژه را باز کردید. پنجره شما شبیه عکس ۱ می شود. البته بر اساس سیستم شما اختلافاتی وجود خواهد داشت.

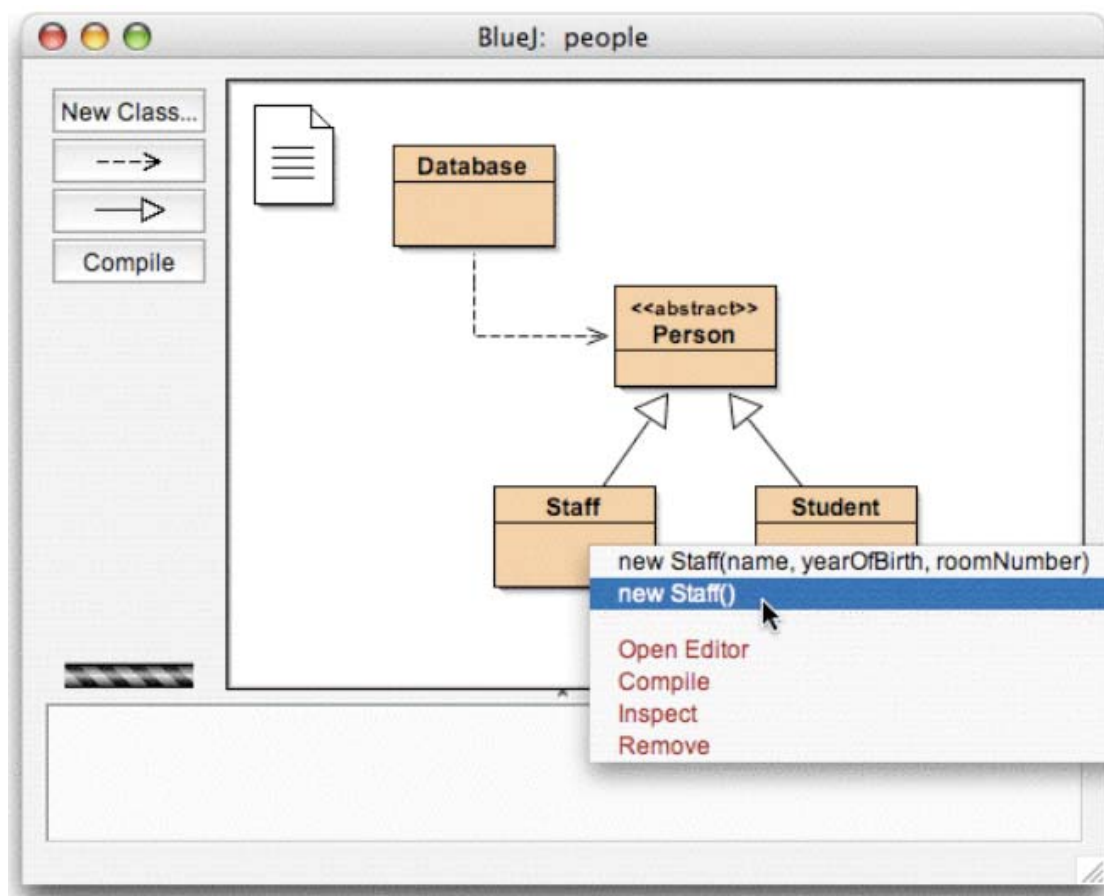
۳-۳) ایجاد اشیا

خلاصه: برای ایجاد یک شی ، سازنده مربوطه را از منوی بازشونده کلاس انتخاب نمایید. یکی از اعمال پایه و ابتدایی در **BlueJ** این است که شما می توانید بدون این که کل پروژه را اجرا نمایید. با یک شی از هر کلاس در ارتباط بوده و متدهای عمومی آنرا اجرا نمایید. مرحله اجرا کردن در **BlueJ** معمولاً با ایجاد یک شی و فراخوانی یکی از متدهای شی انجام می شود. این مورد در طول توسعه یک برنامه می تواند خیلی مفید باشد. شما می توانید کلاس های را که می نویسید. به طور مجزا آزمایش کنید و نیازی به نوشتن کامل برنامه از ابتدا ندارد.

نکته : متدهای **Static** می توانند بدون ایجاد شی از ابتدا اجرا شوند. نمونه ای از این متدهای **static** ، متد **main** میباشد. بنابراین ما می توانیم همچنین کاری را انجام دهیم. همانگونه که در برنامه های کاربردی جاوا اتفاق می افتد. برای اجرا یک برنامه کاربردی ، تنها کافیسیت تا متد **static** ، **main** را اجرا نماییم. ما بعداً به این موضوع بر خواهیم گشت. ابتدا ما قصد داریم تا کارهای جالبی را انجام دهیم که به صورت عادی در جاوا نمی توان انجام داد.

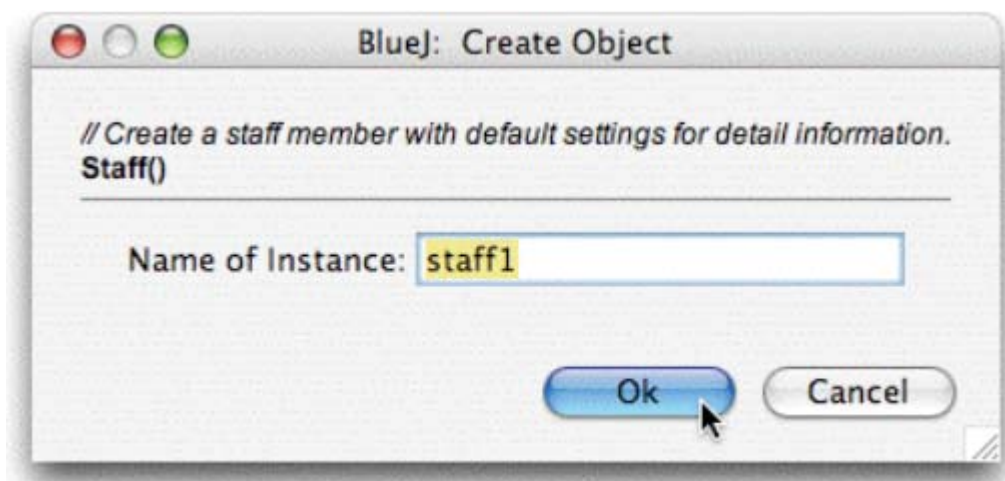
مربع های که شما در وسط پنجره اصلی برنامه می بینید (**Database, person, staff, student**) آیکون های هستند که کلاس های موجود در این پروژه را نمایش می دهند. شما می توانید با کلیک راست کردن (در مکینتاش : **Ctrl-click**) بر روی این آیکون ها منوی از عملیات موجود برای آن کلاس را بدست آورید. (شکل ۲)

این اعمال نمایش داده شده شامل دستور `new` به همراه سازنده های تعریف شده برای کلاس و همچنین دیگر اعمالی که توسط محیط BlueJ فراهم آمده است.



شکل ۲ : اعمال مربوط به کلاس (منوی بازشونده کلاس)

ما قصد داریم تا از کلاس **Staff** یک شی ایجاد نماییم. بنابراین بر روی آیکون **Staff** کلیک راست کرده. (همانند منوی بازشونده شکل ۲). منوی نمایش داده شده شامل دو سازنده برای ساختن شی **Staff** می باشد. یکی با پارامتر و یکی بدون پارامتر. ما در ابتدا سازنده بدون پارامتر را انتخاب می نماییم. پنجره نشان داده شده در شکل ۳ نمایش داده می شود.



شکل ۳ : سازنده شی بدون پارامتر

این پنجره نام شی را که قصد دارید ایجاد کنید می پرسد. ما در این زمان با نام پیش فرض (staff1) موافقت می نماییم. این نام برای حالا مناسب می باشد. شما تنها کافیست دکمه Ok را فشار دهید. حالا شی از کلاس Staff ایجاد می شود. هنگامی که شما شی را ایجاد کردید. شی در قسمت object bench قرار می گیرد (شکل ۴).

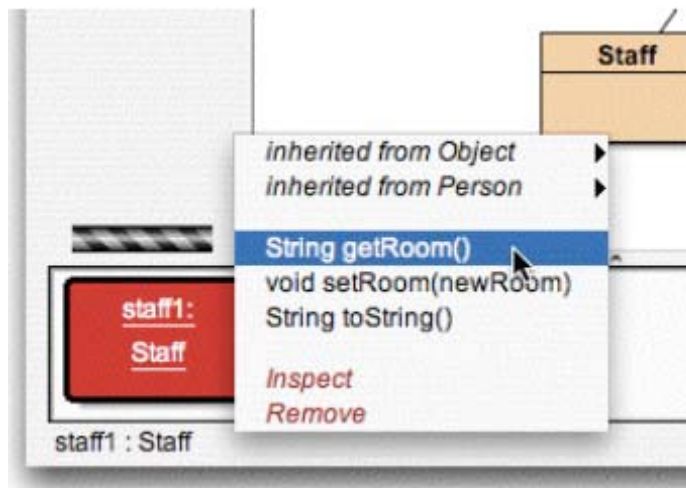


شکل ۴: شی قرار گرفته در قسمت object bench

پس روش ایجاد شی : از منوی بازشونده کلاس سازنده را انتخاب کرده ، آنرا اجرا کنید. حالا شما یک شی در قسمت Object bench خواهید داشت. شما باید دقت کنید که کلاس Person با نشان <<abstract>> علامت گذاری شده (یعنی کلاس abstract هست) شما باید دقت کنید که شما از یک کلاس abstract نمی توانید یک شی ایجاد کنید (همانگونه که در جاوا تعریف شده است)

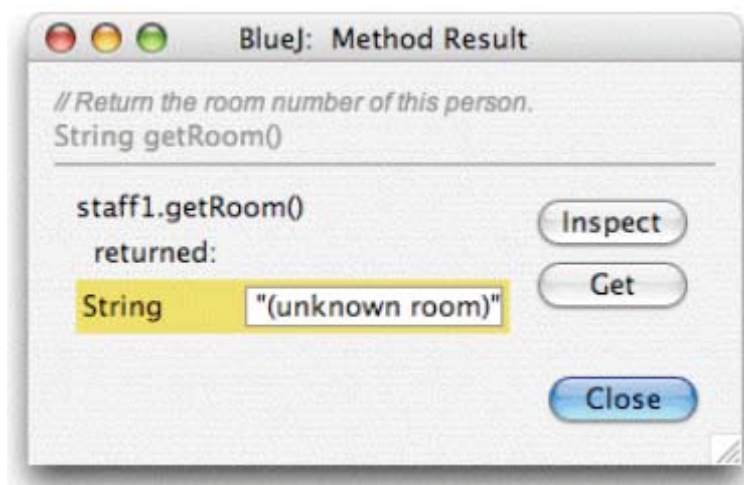
۴-۳) اجرا کردن

خلاصه : برای ایجاد یک شی ، از منوی بازشونده شی متد مربوطه را انتخاب نمایید. حالا که شما یک شی ایجاد کردید. شما می توانید عملیات عمومی را روی آن اجرا نمایید. روی آیکون شی کلیک راست کرده تا منوی بازشونده شی نمایش داده شود (شکل ۵). منوی بازشونده متدهای موجود برای این شی را به همراه دو عملی که توسط محیط BlueJ توسعه یافته اند (Inspect and Remove) را نمایش می دهد. ما بعدا آنها را توضیح خواهیم داد. ابتدا اجازه دهید تا بر روی متدها تمرکز نماییم.



شکل ۵ : منوی شی

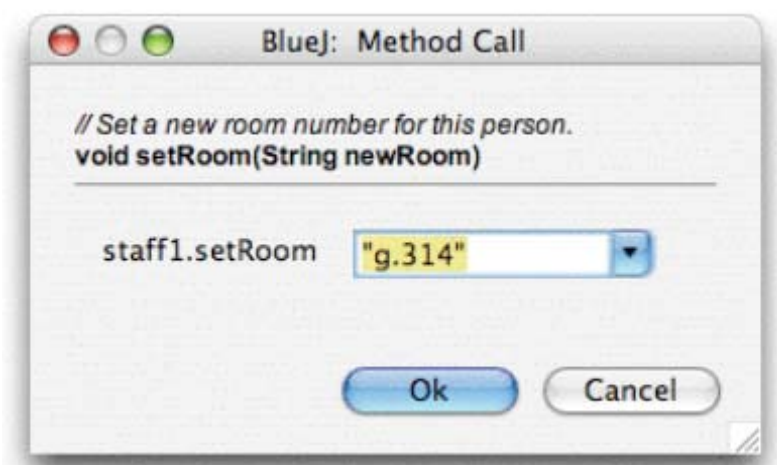
شما در آنجا متدهای `getRoom` و `setRoom` را مشاهده می‌نمایید. که هر کدام تنظیم می‌کند و بر می‌گردانند شماره خانه مربوط به `staff` را. سعی کنید تا متد `getRoom` را فراخوانی نمایید. برای این کار از منوی بازشونده شی متد را انتخاب کنید تا اجرا شود. پنجره ای نمایش داده می‌شود که نتیجه این فراخوانی را به شما نمایش می‌دهد. (شکل ۶) در این مورد؛ نام برگشت داده شده `"unknown room"` می‌باشد. زیرا ما یک نام خاص برای `room` این شخص در نظر نگرفته ایم.



شکل ۶ : نمایش نتیجه تابع

متدهای که از کلاس‌های دیر ارث برده‌اند در زیر منو موجود می‌باشند. در بالای منوی بازشونده شی دو زیر منو موجود می‌باشد. یکی برای متدهای است که از کلاس `Object` ارث برده‌اند. و یکی برای متدهای که از کلاس `Person` شما می‌توانید متدهای کلاس `Person` (مانند `getName`) را با انتخاب آنها از زیر منو فراخوانی نمایید. این کار را انجام دهید. اگر توجه کنید خواهید دید که جواب تولید شده نام مشخص می‌باشد یعنی پاسخ برابر با `"unknown name"` می‌باشد. دلیل آن هم این است که ما به اشخاص نامی نداده ایم. حالا به ما اجازه بدهید تا برای یک اتاق (`room`) یک شماره تعیین کنیم. این کار به ما نشان می‌دهد که چگونه می‌توانیم

یک فراخوانی با پارامتر داشته باشیم. (فراخوانی `getName` و `getRoom` فقط یک مقدار را بر می گرداند. اما پارامتر پارامتر پذیر نیست) تابع `setRoom` را با انتخاب آن از منو فراخوانی نمایید. پنجره ای باز می شود که به شما اجازه می دهد تا پارامتر را وارد نمایید (شکل ۷)



شکل ۷ : پنجره فراخوانی تابع با پارامتر

در بالای پنجره ای که نمایش داده می شود. رابط متد فراخوانی شده نمایش داده می شود. (شامل کامنت و امضا متد) در زیر این قسمت محل ورود متن می باشد که شما با استفاده از آن می توانید پارامتر مربوطه را وارد نمایید. امضا متد که در قسمت بالای پنجره قرار دارد. به ما می گوید که شما یک پارامتر از نوع رشته دارید. یک اتاق جدید از نوع رشته (شامل کوتیشن ها) در محل ورود متن وارد نماید و دکمه `Ok` را فشار دهید. این متد مقدار بازگشتی ندارد و در نتیجه پنجره ای به عنوان نتیجه باز نخواهد شد. بلکه برای این که شما تغییرات را ببینید باید یکبار دیگر متد `getRoom` را فراخوانی نمایید. سعی کنید اشیای را ایجاد کنید و متدهای آنها را فراخوانی نمایید. سعی کنید که سازنده را با آرگومان هایش فراخوانی نمایید. همچنین متدهای مربوط به آنها را فراخوانی نمایید تا با انجام ایم کارها با این اعمال آشنایی بیشتری پیدا نمایید.

۳-۵) ویرایش کلاس

خلاصه : برای ویرایش کد کلاس ، بر روی آیکون کلاس دو بار کلیک نمایید. تا اینجا ، ما فقط با رابط اشیا سرو کار داشتیم. اما حالا وقت اونه که نگاهی به بخش داخلی برنامه داشته باشیم. شما می توانید پیاده سازی یک کلاس را با انتخاب `Open Editor` از توابع کلاس ببینید. (یاد آوری: با کلیک راست کردن بر روی آیکون کلاس می توانید توابع و اعمال مربوط به کلاس را مشاهده نماید) با دوبار کلیک کردن بر روی آیکون کلاس شما به طور میان بر به توابع مشابهی دسترسی خواهید یافت. جزئیات مربوط به استفاده از ادیتور در این آموزش به شرح داده نشده ولی استفاده از آن بسیار سر راست می باشد. جزئیات ادیتور به طور مشخص بعدا توضیح داده خواهد شد. برای حالا ، پیاده سازی مربوط به کلاس `Staff` را باز نمایید. پیاده سازی مربوط به متد `getRoom` را پیدا نمایید. این متد شماره اتاق اعضا عضو را بر می گرداند. به ما اجازه دهید تا با اضافه کردن پیشوند `"room"` به متد نتیجه نهایی تابع را تغییر دهیم (بنابراین متد مقدار `"room M.3.18"` را به جای مقدار `"M.3.18"` بر می

گرداند). ما اینکار را با تغییر این خط برنامه انجام خواهیم داد :

Return room;

و تغییر آن به :

Return "room"+room;

برنامه **BlueJ** بطور کامل ، دستورات جاوا را پشتیبانی می کند. در نتیجه در این جا چیز خاصی برای نحوه پیاده سازی کلاس نیست. و همه چیز بر اساس قواعد جاوا انجام می پذیرد.

۳-۶) کامپایل کردن

خلاصه: برای کامپایل یک کلاس، دکمه **Compile** در ویرایش گر را فشار دهید. و برای کامپایل یک پروژه دکمه **Compile** واقع در پنجره پروژه را فشار دهید.

بعد از اینکه شما متن را وارد کردید (و قبل از این که کار دیگری انجام دهید). نمای پروژه تان را بررسی نمایید (پنجره اصلی) شما اگر دقت کنید خواهید دید که آیکون کلاس **Staff** تغییر کرده است. و آیکون کلاس شامل خطوط راه راهی شده است. این خطوط راه راه نشان می دهند که کلاس بعد از آخرین تغییرات کامپایل نشده است.

توجه : شاید شما تعجب کنید که چرا برای اولین بار که این پروژه را باز کردید آیکون های کلاس خط دار نبودن. این موضوع به این دلیل می باشد که کلاس های پروژه **people** بصورت کامپایل شده توزیع شده اند. البته اکثر پروژه های **BlueJ** بصورت کامپایل نشده توزیع می شوند. بنابراین از این به بعد، به آیکون کلاس های پروژه های که باز می کنید دقت نمایید.

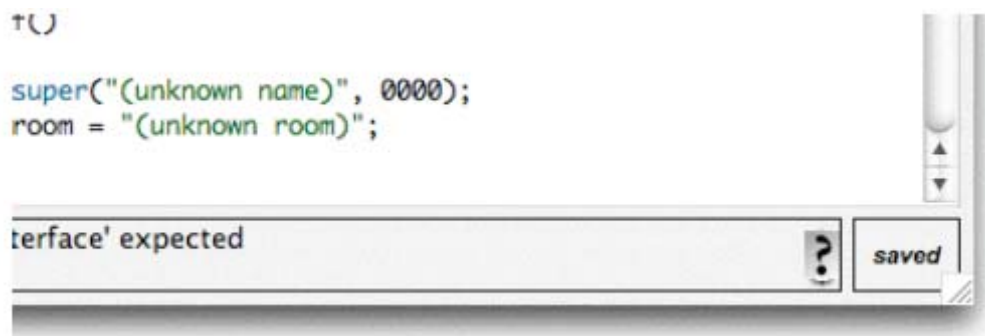
در منوی ابزار واقع در بالای ویرایشگر چندین دکمه قرار دارند که ما از آنها مرتباً استفاده می کنیم. یکی از این دکمه ها دکمه **Compile** می باشد. این تابع به شما اجازه می دهد تا کلاس تان را مستقیماً از درون ویرایشگر کامپایل نمایید. حالا دکمه **Compile** را فشار دهید. اگر شما اشتباهی مرتکب نشده باشید. پیغام "**the class has been compiled**" در بخش اطلاعات واقع در انتهای ویرایشگر به نمایش در می آید. اگر شما خطایی انجام دهید که منجر به تولید یک خطای نحوی شود. خط کدی که خطا در آن اتفاق افتاده مشخص () می شود و همچنین پیغام خطایی در بخش اطلاعات برنامه به نمایش در می آید. (در مورد اولی که کامپایل کردید. سعی کنید یک خطای نحوی ایجاد کنید به عنوان مثال یک سیمی کلون را حذف کنید. حالا آنرا مجدداً کامپایل نمایید تا ببینید چه اتفاقی می افتد.) بعد از این که کلاس را با موفقیت کامپایل نمودید. ویرایشگر را ببندید.

توجه : شما نیاز به ذخیره کد کلاس ندارید کدها در زمان های مورد نیاز به صورت خودکار ذخیره می شوند. (مانند هنگامی که ویرایشگر را می بندید و یا قبل از این که کلاس را کامپایل نمایید. شما اگر دوست دارید می توانید کدها را ذخیره کنید (تابع مورد نیاز در ویرایشگر مربوط به کلاس وجود دارد) البته شما در حالتی به این کار نیاز دارید که کامپایلر تان به صورت مکرر کرش (crash) می کند. و شما نگرانید که کارتان از بین برود.

منوی ابزار پنجره پروژه نیز شامل دکمه Compile می باشد. با فشردن این دکمه کل پروژه کامپایل می شود. (این دکمه تشخیص می دهد که کدامیک از کلاس ها نیاز به کامپایل مجدد دارند. و سپس بعد از کامپایل مجدد آنها را مرتب می کند). این کار را با ایجاد تغییر در دو یا چند کلاس (بنابراین شما باید دو یا چند کلاس در صفحه ترسیم کلاس ها داشته باشید که با خطوط راه راه مشخص شده باشد) امتحان کنید و سپس دکمه Compile را فشار دهید. اگر در حین کامپایل خطایی در یکی از کلاس ها تشخیص داده شود. ویرایشگر باز شده و محل خطا و همچنین پیغام مناسب را نمایش می دهد. شما اگر دقت کنید خواهید دید که محل اشیا (object bench) مجدداً خالی می شود. هنگامی که در پیاده سازی تغییر ایجاد شود اشیا پاک می شوند. چ

۷-۳ راهنمایی بوسیله خطاهای کامپایلر

خلاصه : برای راهنمایی گرفتن درباره پیغام خطایی کامپایلر ، بر روی علامت سوال که بعد از پیغام خطا قرار دارد کلیک کنید. اغلب اوقات ، دانشجویان مبتدی به سختی متوجه پیغام های خطا می شوند. ما تلاش کردیم تا یک راهنما نیز ایجاد نماییم. ویرایشگر را مجدداً باز کنید و در آن یک خطا ایجاد کرده و کلاس را دوباره کامپایل نمایید. در بخش اطلاعات ویرایشگر باید پیغام خطا نمایش داده شود. در انتهای سمت راست بخش اطلاعات علامت سوال مشخص می باشد که شما با کلیک کردن بر روی آن می توانید اطلاعات بیشتری درباره آن خطا بدست آورید. (عکس شماره ۸)



عکس ۸ : خطای کامپایلر و دکمه راهنما

در این مرحله ، متن های راهنما برای تمام پیغام های خطا موجود نمی باشد. مقداری از متن های راهنما هنوز در حال نوشته شدن می باشند. اما این خاصیت ارزش امتحان کردن را دارد. تعداد زیادی از خطا ها توضیح داده شده اند. موارد باقی مانده نیز نگاشته شده و در ویرایش های بعدی BlueJ به آن اضافه خواهند شد.

فصل ۴- انجام اعمالی بیشتر

در این فصل ما بیشتر بر روی کارهای که شما می‌توانید توسط محیط BlueJ انجام دهید. بحث خواهیم کرد این موارد پایه‌ای نیستند ولی به طور فراوان مورد استفاده قرار می‌گیرند.

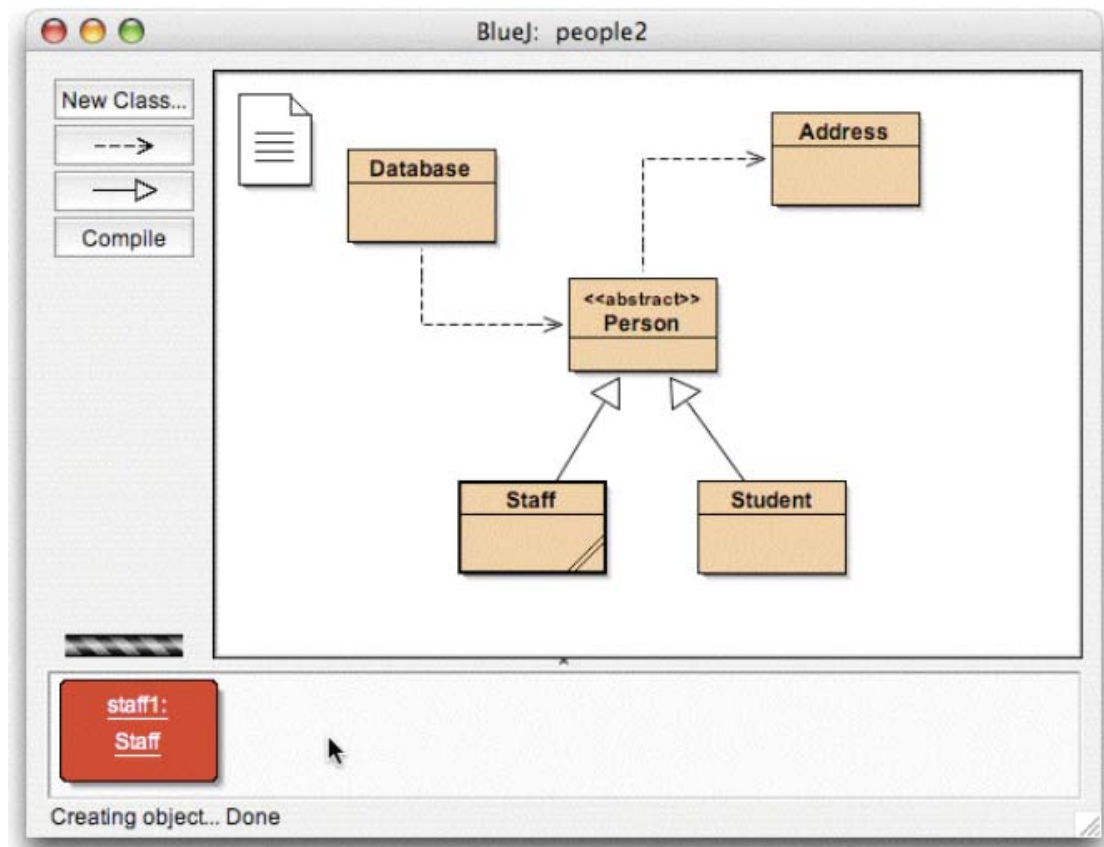
۴-۱) ناظر

خلاصه: ناظر شی به شما اجازه می‌دهد تا به رفع خطاهای ساده بپردازید با نشان دادن وضعیت داخلی شی. هنگامی که شما متدهای یک شی را اجرا می‌کنید. شما می‌بینید که تابع **Inspect** برای اشیای که شامل متدهای تعریف شده توسط کاربر هستند موجود است (شکل ۵). این تابع به شما اجازه می‌دهد تا وضعیت متغیرهای (فیلدها) شی را بررسی نمایید. یک شی با چند مقدار تعریفی توسط کاربر ایجاد نمایید (به عنوان مثال شی **Staff** با سازنده‌های که پارامتر می‌پذیرند). سپس تابع **Inspect** را از منوی شی انتخاب نمایید. پنجره‌ای نمایش داده می‌شود که فیلد‌های شی را با نوع و مقدارشان نمایش می‌دهد (شکل ۹).



شکل ۹: پنجره ناظر

ناظر برای بررسی سریع اعمالی که تغییر می‌کنند بسیار مفید می‌باشد (اعمالی که موقعیت اشیاء در آن تغییر می‌نماید) بنابراین، ناظر یک ابزار رفع خطا ساده می‌باشد. در مثال **Staff**، همه فیلدها از نوع ساده می‌باشند (غیر شی یا رشته‌ها). مقدار این انواع می‌تواند مستقیماً نمایش داده شود. شما می‌توانید فوراً ببینید که کدام سازنده شامل متغیر نهایی می‌باشد. در بسیاری از نمونه‌های پیچیده، مقادیر فیلدها می‌تواند به اشیاء که توسط کاربران تعریف کرده‌اند مراجعه کند. برای مشاهده چنین حالتی ما باید از یک پروژه دیگر استفاده کنیم. پروژه **people2** را باز کنید. این پروژه همراه توزیع استاندارد **BlueJ** موجود می‌باشد. بعد از باز کردن پروژه **people2** نمای پنجره پروژه شبیه شکل ۱۰ خواهد شد. در مثال دوم شما می‌توانید ببینید که کلاس **Address** به کلاس‌های مثال قبلی اضافه شده است. یکی از فیلدهای کلاس **Person** از نوع تعریف کاربر کلاس **Address** می‌باشد.

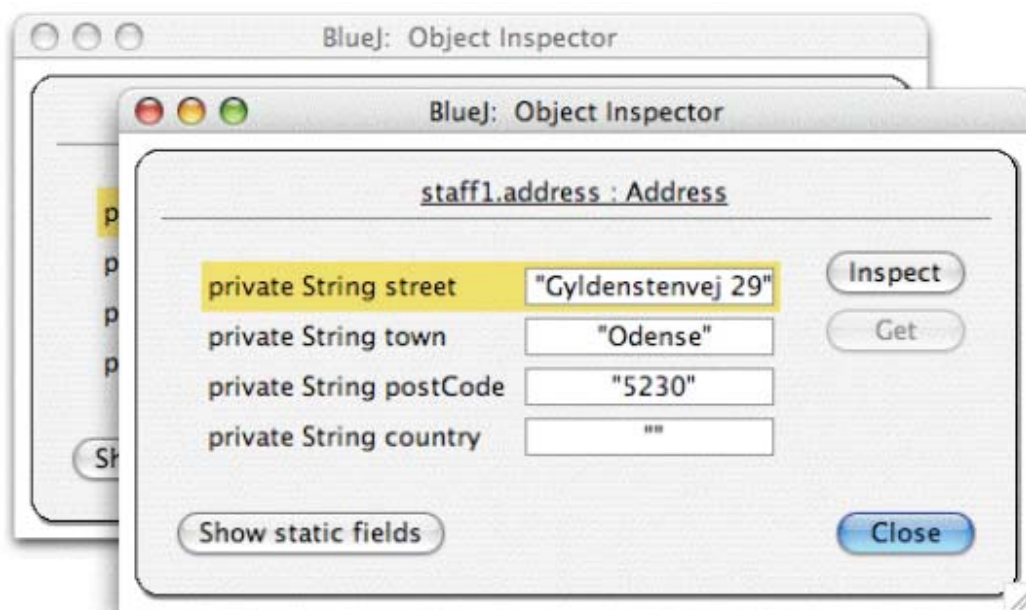


شکل ۱۰: پنجره پروژه people2

کار بعدی که قرار است کا انجام دهیم . ناظر با فیلدهای نوع شی می باشد. یک شی Staff ایجاد کنید و سپس متد setAddress() این شی را فراخوانی نمایید (شما این متد را می توانید در زیر منوی person پیدا نمایید). آدرس را وارد نمایید. به طور داخلی ، کد Staff شی کلاس Address را ایجاد می کند. و آنرا در فیلد address ذخیره می نماید. حالا ، ناظر شی Staff را فراخوانی نماید. شکل شماره ۱۱ نتیجه را به ما نشان می دهد. حالا فیلدهای درون شی Staff شامل address می باشند. شما همچنین می توانید ببینید که مقدار این فیلد با یک پیکان نمایش داده شده که این بدین معنی است که مقدار آن به یک شی دیگر مراجعه می نماید. خب تا اینجا این موضوع پیچیده می باشد. زیرا مقدار شی های تعریفی توسط کاربر را نمی توان مستقیما در این لیست مشاهده کرد. برای امتحان کردن مقادیر مربوط به address ، فیلد Address را در لیست انتخاب کرده و دکمه Inspect مربوط به پنجره را فشار دهید (شما همچنین می توانید بر روی فیلد address دو بار کلیک کنید) پنجره ناظر دیگری باز شده و جزئیات شی Address را نمایش می دهد (شکل ۱۲).



شکل ۱۱: ناظر با مراجعه به شی



شکل ۱۲: ناظر شی داخلی

اگر فیلد انتخاب شده از نوع **public** میباشد شما به جای فشار دادن دکمه **Inspect** می توانید ، شما می توانید فیلد **Address** را انتخاب نمایید و دکمه **Get** را فشار دهید. با این عمل شی انتخاب شده در **object bench** مشخص می شود. شما در اینجا می توانید با فراخوانی متدهای شی به آزمایش خود ادامه دهید.

۲-۴) ارسال اشیا به عنوان پارامتر

خلاصه: یک شی را می توان به عنوان پارامتر برای فراخوانی یک متد ارسال کرد. برای اینکار بر روی آیکن شی کلیک کنید.

اشیا می توانند به عنوان پارامترها به متدهای پروژه های دیگر ارسال شوند. اجازه دهید تا با یک مثال این کار را انجام دهیم

یک شی از کلاس Database ایجاد نمایید (شما باید توجه نمایید که کلاس Database شامل یک سازنده می باشد که پارامتری را دریافت نمی کند بنابراین ساخت شی سر راست میباشد) شی Database قادر است تا لیستی از اشخاص را در خود نگه دارد.

این شی شامل اعمالی برای اضافه کردن اشیا شخص و همچنین نمایش شخص های که تا کنون ذخیره شده اند می باشد.

(در حقیقت نام Database یک اغراق کوچک می باشد !!!) اگر شما در حال حاضر شی Staff و یا Student را در Object bench ندارید. یکی از این دو شی را ایجاد نمایید. برای ادامه شما نیاز دارید تا شی Database و شی Staff و یا Student در یک زمان در object bench قرار داشته باشند. حالا متد addPerson را از شی Database فراخوانی نمایید امضا متد به شما می گوید که پارامتر نوع Person مورد انتظار می باشد. (یاد آوری: کلاس Person از نوع انتزاعی (abstract) می باشد بنابراین انجا اشیا می که مستقیما از نوع person باشند نیست. اما یکی از زیر انواع شی Staff و یا Student می تواند جایگزین مناسبی برای شی Person باشد. بنابراین این قانونی است که ما اشیا Staff و یا Student را به عنوان پارامتر ارسال کنیم هر جا که Person مورد انتظار می باشد). برای ارسال شی به عنوان پارامتر بر روی شی موجود در object bench کلیک کرده و از پنجره باز شده نام کلاس مورد نظر را در فیلدهای مربوطه وارد می نماییم. دکمه Ok را فشار دهید تا فراخوانی مربوطه انجام پذیرد. از آنجایی که این متد مقدار بازگشتی ندارد ما نمی توانیم فوراً نتیجه را مشاهده نماییم. شما می توانید متد listAll را از شی Database فراخوانی نمایید تا بررسی کنید که این عمل برای شی انجام شده است. دستور listAll اطلاعات شخص را در خروجی استاندارد چاپ می کند شما باید توجه کنید که صفحه متن () به طور خودکار باز شده و متن را نمایش می دهد.

این کار را با وارد کردن اشخاص بیشتری در Database تمرین نمایید.

فصل ۵- ایجاد پروژه جدید

این فصل به شما یک آموزش سریع برای تنظیم یک پروژه جدید می دهد.

۱-۵) ایجاد پروژه جدید

خلاصه: برای ایجاد یک پروژه از منوی **Project** گزینه **New....** را انتخاب نمایید.

برای ایجاد یک پروژه جدید از منوی **Project** گزینه **New....** را انتخاب نمایید. پنجره انتخاب فایل باز شده و به شما اجازه می دهد تا نام و محل ذخیره سازی پروژه جدید را مشخص نمایید. این کار را حالا امتحان نمایید. شما می توانید هر گونه نامی برای پروژه تان انتخاب نمایید. بعد از اینکه شما دکمه **Ok** را فشار دادید. در مسیر مورد نظر پوشه ای همان نامی که شما انتخاب کرده اید ایجاد می شود. و پنجره اصلی برنامه بصورت خالی نمایش داده می شود.

۲-۵) ایجاد کلاسها

خلاصه: برای ایجاد یک کلاس ، دکمه **New Class** را فشار می دهیم. و برای کلاس یک نام قرار می دهیم.

شما اکنون می توانید با کلیک کردن بر روی دکمه **New Class** واقع در منوی ابزار یک کلاس جدید ایجاد کنید. شما می توانید سوال کنید برای تهیه کردن یک نام برای کلاس- این نام می تواند یک شناسه معتبر جاوا باشد.

شما همچنین می توانید ۴ نوع کلاس را انتخاب نمایید : **standard** ، **applet** ، **abstract** ، **interface**

این انتخاب مشخص می کند که بدنه اولیه کدهای که برایتان تولید می شود چگونه باشد. شما بعدا می توانید نوع کلاس را با تغییر در کدهای آن ویرایش کنید (به عنوان مثال ، با اضافه کردن کلمه کلیدی **abstract** در بین کدها) .

بعد از این که شما کلاس را ایجاد نمودید. کلاس به وسیله یک آیکون در صفحه نمایش ، نمایش داده می شود. اگر کلاس شما از نوع استاندارد نیست (**abstract, applet, interface**) این موضوع در آیکون کلاس نمایش داده خواهد شد.

هنگامی که ویرایشگر را برای کلاس جدید باز می کنید خواهید دید که کدهای اولیه برای این کلاس ایجاد شده است.

این موضوع ، کار شما برای شروع را آسان تر خواهد کرد. کدهای پیش فرض از لحاظ نحوی صحیح می باشد. شما می توانید آنرا کامپایل نمایید (اما این کلاس ها کاری انجام نمی دهند) . سعی کنید چند کلاس ایجاد کنید و آنها را کامپایل نمایید.

۳-۵) ایجاد وابستگی ها

خلاصه: برای ایجاد یک پیکان ، بر روی دکمه پیکان کلیک کرده و آنرا بر روی صفحه ترسیم درگ نماید. و یا فقط در ویرایشگر کد مربوطه را تایپ نماید.

در صفحه ترسیم کلاس ها ، علامت پیکان روابط ما بین کلاس ها را مشخص می نماید.

روابط ارث بری (**Implement** ، **extend**) بوسیله پیکانی با سر تو خالی (سر بسته) مشخص می شود. روابط **"uses"** با یک پیکان نقطه چین با سر باز مشخص می شود.

شما می توانید روابط را مستقیما در صفحه ترسیم و یا بوسیله نوشتن کد در ویرایشگر مشخص نماید. شما اگر وابستگی ها را بوسیله گرافیکی ایجاد کنید. کدها بصورت خودکار بروز رسانی می شوند و اگر شما وابستگی ها را بوسیله کدها ایجاد کنید. بخش گرافیکی به صورت خودکار بروز رسانی می شود.

برای اضافه کردن یک پیکان بصورت گرافیکی ، بر روی دکمه پیکان مقتضی(پیکان سر بسته برای روابط ارث بری و پیکان نقطه جین برای uses) کلیک کرده و آنرا از بین دو کلاس مربوطه درگ می نماییم.

با اضافه کردن یک پیکان مربوط به ارث بری کد توضیحی مربوط به extend و یا implement به کدهای کلاس تان اضافه می شود. (این بستگی به مقصد دارد که از نوع کلاس باشد و یا interface).

اضافه کردن پیکان uses بلافاصله تغییری در کدها ایجاد نمی کند (مگر اینکه مقصد کلاسی در یک پکیج دیگر باشد، در این نمونه دستور Import تولید می شود. اما ما فعلا آنرا در بین مثال هایمان نداریم). استفاده از پیکان uses برای کلاسی که مورد استفاده قرار نگرفته در حین کامپایل خطاری ایجاد می کند مبنی بر این که شما از ارتباط uses بین کلاس ها تعریف شده ولی کلاس مورد استفاده قرار نگرفته است.

اضافه کردن پیکان ها برای ارتباط در حالت متنی بسیار ساده می باشد: شما کافی است که به صورت معمولی کدها را تایپ نماید. هنگامی که کلاس را ذخیره کنید. صفحه ترسیم بروز رسانی می شود(یادآوری : با بستن ویرایشگر ، کلاس به صورت خودکار ذخیره می شود).

۴-۵ حذف کردن عناصر

خلاصه : برای حذف یک کلاس و یا یک پیکان ، از منوی باز شونده گزینه remove را انتخاب نماید.

برای حذف یک کلاس از صفحه ترسیم ، کلاس را انتخاب نماید و از منوی Edit ، گزینه remove را انتخاب نماید. شما همچنین می توانید از منوی باز شونده کلاس گزینه remove را انتخاب نمایید. هر دو مورد توضیح داده شده برای کلاس برای یک پیکان نیز کاربرد دارد.

فصل ۶ – استفاده از code pad

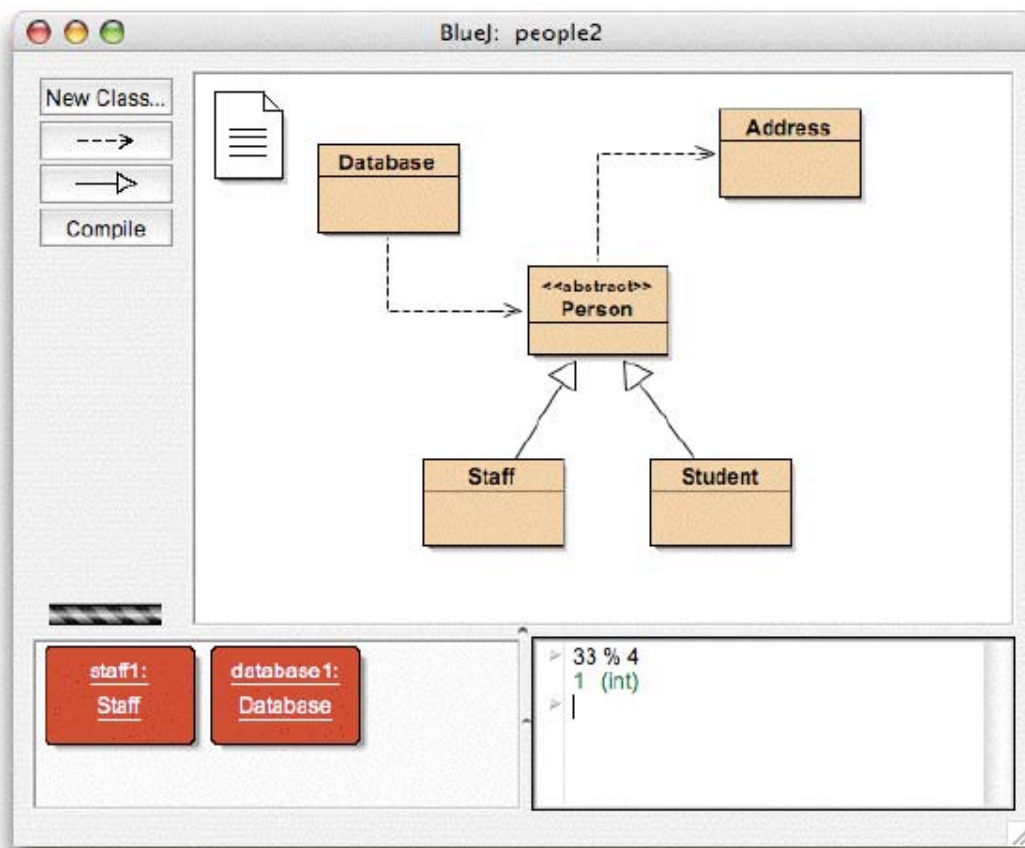
با استفاده از Code Pad محیط BlueJ شما می توانید با سرعت و به سادگی به آزمایش دستورات و عبارات یک تکه کد جاوا بپردازید. بنابراین ، کد پد می تواند برای بررسی دستورات و قواعد زبان جاوا مورد استفاده قرار بگیرد.

۶-۱) نمایش کد پد

خلاصه : برای نمایش کد پد از منوی View گزینه Show Code Pad را فشار دهید.

کد پد بصورت پیش رض نمایش داده نمی شود. برای دیدن آن از منوی View گزینه Show Code Pad را فشار دهید.

پنجره اصلی حالا شامل کد پد در گوشه سمت راست و بعد از object bench خود می باشد (شکل ۱۳)



شکل ۱۳ : پنجره اصلی به همراه کد پد

اندازه پنجره کد پد و همچنین Object Bench با استفاده از خطوط عمودی و افقی قابل تغییر می باشد. کد پد حالا می تواند برای ورود دستورات و عبارات مورد استفاده قرار بگیرد. با فشار دادن دکمه Enter هر خط اجرا شده و نتیجه آن نمایش داده می شود.

۶-۲) اجرای یک عبارت ساده

خلاصه : برای اجرای یک عبارت جاوا ؛ تنها کافیست که انرا در کد پد تایپ نمایید.

کد پد می تواند برای اجرا و امتحان کردن عبارات ساده بکار برود. به عنوان مثال ، عبارات زیر را امتحان نمایید:

```

4 + 45
"hello".length()
Math.max(33, 4)
(int) 33.7
javax.swing.JOptionPane.showInputDialog(null, "Name:")

```

عبارات می توانند به مقدارهای استاندارد جاوا مراجعه کنند و اشیا می توانند به کلاس های موجود در پروژه مراجعه داشته باشند. کد پد نتایج را نمایش خواهد داد و اگر عبارت شما اشتباه باشد پیام خطایی نشان خواهد داد.

شما می توانید از اشیا که در **object bench** قرار دارند نیز استفاده کنید. برای نمونه: یک شی از کلاس **student** در **Object bench** قرار دهید (با استفاده از منوی بازشونده کلاس که قبلا توضیح داده شد). نام شی **student1** هست. حالا شما در محیط کد پد تایپ نمایید:

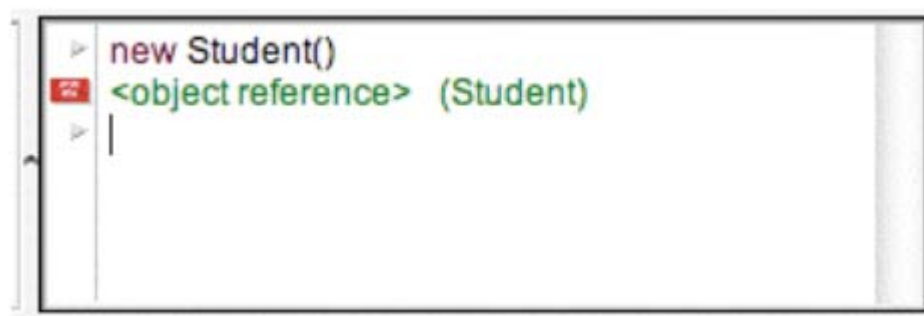
```
student1.getName()
```

به طور مشابه، شما می توانید به تمام متدهای در دسترس کلاس های پروژه تان دسترسی پیدا نمایید.

۳-۶ دریافت اشیا

خلاصه: برای انتقال اشیا از کد پد به محیط **object bench**، آیکون کوچک شی را به آن محیط درگ کنید.

نتیجه بعضی از عبارات (دستورات) یک شی می باشد. که شامل مقدارهای ساده ای هستند. در این مثال، نتیجه نمایش داده شده **<object reference>** یک نوع شی می باشد. که به وسیله آیکونی که در کنار آن ترسیم شده در خط بعد از عبارت مشخص می باشد.



شکل ۱۴: نتیجه یک عبارت در کد پد، یک شی می باشد

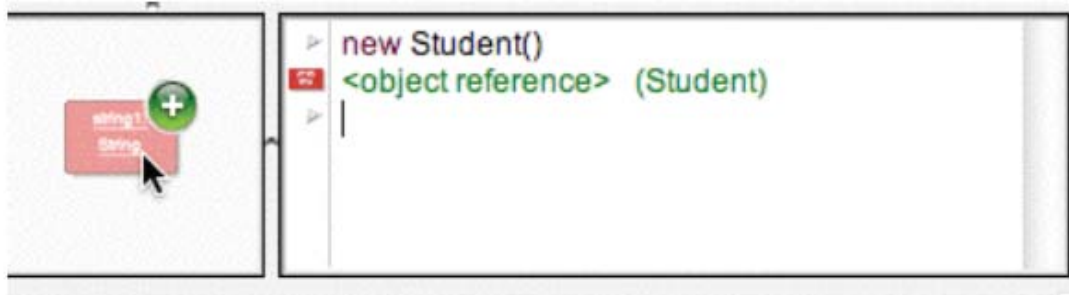
اگر نتیجه یک رشته باشد شما مقدار انرا می توانید ببیند. در ضمن باز هم یک آیکون در کنار آن ترسیم شده است (زیرا رشته ها نیز شی هستند). تعدادی از دستوراتی که شما با استفاده از آن ها می توانید شی ایجاد نمایید:

```

new Student()
"marmelade".substring(3,8)
new java.util.Random()
"hello" + "world"

```

شما می توانید آیکون کوچک شی را درون **Object Bench** درگ کنید تا با استفاده از شی نتیجه شده کار را ادامه دهید.



شکل ۱۵ : درگ کردن شی درون bench object

۴-۶) بررسی اشیا

خلاصه : برای بررسی نتیجه اشیا در کد پد بر روی آیکن کوچک شی دوبار کلیک نمایید.
 اگر شما می خواهید یک شی را که به عنوان نتیجه برگشت داده شده را از طریق کد پد بررسی نماید بدون اینکه شی را در Object bench قرار دهید. می توانید بر روی آیکن شی دوبار کلیک کرده تا ناظر شی باز شود.

۵-۶) اجرا کردن دستورات

خلاصه : دستوراتی که در درون کد پد تایپ می شوند. اجرا می گردند
 شما همچنین می توانید از کد پد برای اجرا کردن دستورات استفاده نمایید (این موضوع برای دستورات جاوای می باشد که مقداری را بر نمی گرداند). این کار را انجام دهید. به عنوان مثال :

```
System.out.println("Gurkensalat");
System.out.println(new java.util.Random().nextInt(10));
```

دستوراتی که اجرا می شوند به درستی اجرا می شوند خواه در انتهای آنها سیمی کالون باشد یا نه.

۶-۶) دستورات چند خطی و دستورات توالی دار

خلاصه : برای دستورات چند خطی در انتهای هر خط از Enter-shift استفاده نمایید.
 شما می توانید توالی از دستورات و یا دستورات چند خطی را با استفاده از Enter-shift در انتهای هر خط ایجاد نمایید.
 با فشردن enter-shift نشانگر موس به ابتدای خط بعدی منتقل می شود بدون این که خط قبلی اجرا شود. در انتهای خط آخر Enter را فشار دهید تا کل خط های با هم اجرا شوند. برای نمونه کد زیر را اجرا نمایید:

```
for (int i=0; i<5; i++) {
System.out.println("number: " + i);
}
```

۷-۶) کار با متغیرها

خلاصه : متغیرهای محلی را می توان در دستورات یک خطی و یا چند خطی مورد استفاده قرار داد. اسامی اشیا موجود در **object bench** در فیلدهای شی به کار گرفته می شوند.

متغیرها (محلی و یا شی) می توانند به صورت محدود در کد پد مورد استفاده قرار بگیرند. شما می توانید متغیرهای محلی را در کد پد تعریف نمایید- اما این فقط برای بخشی از دستورات توالی چند خطی مناسب می باشد. برای مثال شما می توانید بلوک دستورات زیر را وارد نمایید و این دستورات درست کار کنند :

```
int sum;
sum = 0;
for (int i=0; i<100; i++) {
sum += i;
}
System.out.println("The sum is: " + sum);
```

شما می توانید با درگ کردن شی از کد پد به **object bench** یک فیلد شی جدید ایجاد نمایید.

۸-۶ تاریخچه دستورات (command history)

خلاصه: با استفاده از دکمه های بالا و پایین روی صفحه کلید می توانید به تاریخچه ورودی ها در کد پد دست پیدا کنید.

کد پد سابقه ای از ورودی های قبلی شما را نگه داری می کند. شما با استفاده از دکمه های بالا و پایین روی صفحه کلید می توانید به خط های ورودی قبلی دست پیدا کنید و می توانید قبل از اجرا مجدد آنها به ویرایش آنها پردازید.

این بخش مقدمه ای است بر مهمترین اعمال مربوط به رفع خطا در BlueJ .

بسیاری از اساتید علوم کامپیوتر معتقدند که استفاده از نرم افزارهای رفع خطا (Debugger) در سال اول آموزش می تواند جالب و مفید باشد. اما فرصتی برای معرفی چنین چیزی را به دانشجویان ندارند. دانشجویان به سختی در تقلا کار کردن با ویرایشگر ، کامپایلر و اجرا کننده هستند . و فرصتی برای آشنایی با یک ابزار پیچیده دیگر را ندارند. بر همین اساس ما مصمم شدیم تا یک دیباگر ساده طراحی کنیم. هدف این بود که دیباگر تهیه شود که شما ظرف ۱۵ دقیقه کل خصوصیات آن را شرح دهید. و دانشجویان بتوانند بدون آموزش خاصی از آن استفاده نمایند. حال ببینید که آیا ما موفق بوده ایم.

در ابتدا امر ، ما اعمال دیباگر های قدیمی را به ۳ وظیفه کاهش دادیم :

۱. تنظیم breakpoint

۲. کامپایل خط به خط کدها

۳. نظارت بر متغیرها (بررسی متغیرها)

در واقع این ۳ عمل بسیار ساده می باشند. در ادامه ما با آنها کار خواهیم کرد.

برای شروع کار ، پروژه debug demo را از پوشه examples انتخاب نمایید. این پروژه شامل کلاس های برای آزمایش کردن توابع و اعمال دیباگر می باشند. و نمی توانند کار دیگری انجام دهند.

۷-۱) تنظیم break point ها :

خلاصه: برای قرار دادن یک break point ، در محل مربوط به break point ها در سمت چپ محیط کدنویسی کلیک کنید.

با تنظیم یک break point شما می توانید در اجرای برنامه در این نقطه معین از کد وقفه ایجاد کنید. و برنامه در آن قسمت متوقف شود. هنگامی که اجرا برنامه متوقف شد . شما می توانید وضعیت اشیا برنامه خود را بررسی کنید. این کار در اغلب اوقات به شما کمک می کند. تا بتوانید بفهمید که در کدهای شما چه اتفاقی می افتد.

همانگونه که در شکل ۱۶ مشخص می باشد. سمت چپ محیط کدنویسی محل مربوط به break point ها می باشد.

شما با کلیک کردن در این محیط می توانید یک break point ایجاد نمایید. علامت کوچک " stop " نقطه

Break point شما را مشخص می نماید. می توانید همین حالا امتحان نمایید.

کلاس Demo را باز کنید. متد loop را پیدا نماید. و نقطه breakpoint خود را در جایی از متد که شامل حلقه

For می باشد. قرار دهید. علامت " stop " در محیط کدنویسی نمایش داده خواهد شد.

```

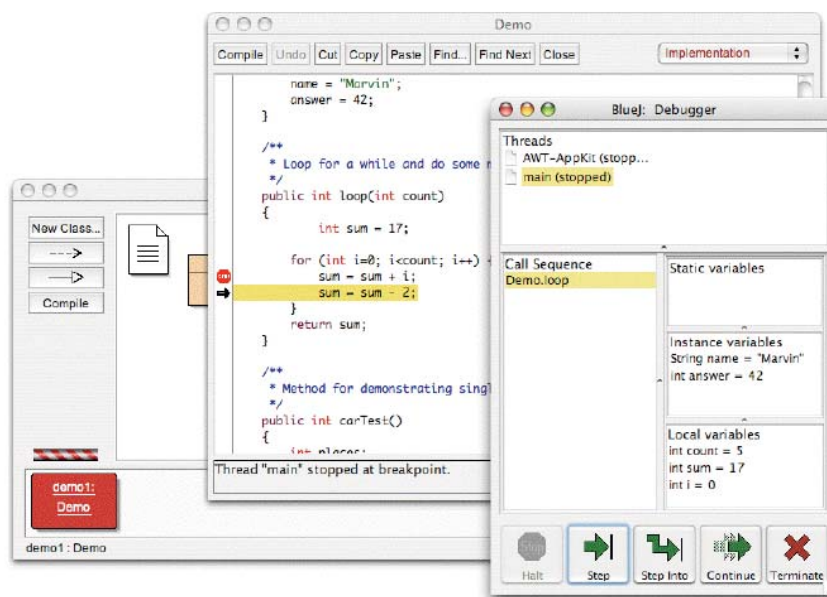
public int loop(int count)
{
    int sum = 17;

    for (int i=0; i<count; i++) {
        sum = sum + i;
        sum = sum - 2;
    }
    return sum;
}

```

شکل ۱۶ : یک نقطه break point

هنگامی که خطوط برنامه را بوسیله نقاط break point مشخص کردید. اجرا برنامه در آن قسمت ها متوقف می شود. اجازه بدید این کار رو انجام بدهیم. شی ای از کلاس Demo ایجاد کنید. و متد Loop آنرا با پارامتر ۱۰ فراخوانی نمایید. به زودی نقطه break point اثر کرده. محیط کد نویسی برنامه باز می شود. و خط جاری را نمایش می دهد. همچنین دیباگر برنامه نیز باز می شود. این وضعیت را می توانید در شکل ۱۷ ببینید.



شکل ۱۷ : پنجره دیباگر برنامه

خطی از برنامه که در محیط کدنویسی مشخص (high Light) شده ، خطی است که قرار است اجرا شود. یعنی برنامه در خط قبلی متوقف شده است.

۲-۷) اجرا خط به خط کدها

برای اجرا خط به خط کدها یثان دکمه **Step and Step Into** را در دیباگر فشار دهید. حال که ما برنامه را متوقف کردیم. می توانیم اجرا یک مرحله از کدها را مشاهده کنیم. و ببینیم که در حین اجرا چه اتفاقی می افتد. برای انجام این کار ، بصورت تکراری ، دکمه **Step** را از پنجره دیباگر فشار دهید. شما باید تغییراتی در خطوط برنامه تان ببینید (خط واضح شده به خطی که اجرا شده منتقل می شود) هر وقت که شما دکمه **Step** را فشار دهید. یک خط از برنامه اجرا شده و اجرا برنامه متوقف می شود. توجه کنید که مقدار متغیرهای درون پنجره دیباگر نیز تغییر می کند (به عنوان مثال مقدار متغیر **sum**) بنابراین شما می توانید مرحله به مرحله برنامه تان را اجرا کنید و ببینید که چه اتفاقی می افتد. برای این که به این کار خاتمه دهید. بر روی نقاط **breakpoint** کلیک کنید. تا پاک شوند و همچنین دکمه **Continue** موجود در دیباگر را فشار دهید تا اجرا برنامه بصورت عادی پیگیری شود. اجازه دهید تا این مراحل را با یک متد دیگر امتحان نماییم. نقطه **Breakpoint** را بر روی خط

```
Places = myCar.seats ( );
```

مربوط به متد **carTest()** از کلاس **Demo** تنظیم می نمایم.

۳-۷) بررسی متغیرها

خلاصه : بررسی متغیرها بسیار ساده می باشند. متغیرها بصورت خودکار در دیباگر به نمایش در می آیند. هنگامی که شما کدهایتان را رفع خطا (**debug**) می نماید. بسیار مهم می باشد که قادر باشید. از وضعیت اشیا باخبر باشید (مثلا : متغیرهای محلی و متغیرهای نمونه) . انجام این کار بسیار ساده و جزئی می باشد. شما نیاز به یک دستور خاص برای بررسی متغیرها ندارید. متغیرهای استاتیک ، متغیرهای نمونه از شی جاری ، متغیرهای محلی متد جاری همواره به صورت خودکار نمایش داده شده و همچنین بروز رسانی نیز می شوند. شما می توانید متدها را در فراخوانی متوالی انتخاب کنید. تا متغیرهای دیگر اشیا و متدهای فعال جاری را مشاهده نمایید. به عنوان مثال ، مجددا یک نقطه **break point** در متد **carTest()** قرار دهید. در سمت چپ پنجره دیباگر شما فراخوانی متوالی (**call sequence**) را مشاهده خواهید کرد. که در حال حاضر موارد زیر را نمایش می دهد:

```
Car.seats  
Demo.carTest
```

این اشاره می کند به این که **Car.Seats** بوسیله **Demo.carTest** فراخوانی شده است. شما می توانید از این لیست گزینه **Demo.carTest** را برای بررسی کد مربوطه و همچنین مقدار متغیرهای جاری این متد انتخاب نمایید. اگر شما از خطی که شامل ساختار **new Car(...)** می باشد. عبور کنید می توانید مقدار متغیر محلی **my car** را که بصورت **<object refrence>** می باشد مشاهده نمایید. مقدار تمام انواع اشیا (به جز رشته ها) از این راه نمایش داده می شود. شما می توانید با دو بار کلیک کردن بر روی متغیرها آنها را بررسی نمایید. انجام این کار شبیه باز کردن پنجره بررسی شی می باشد . که در بخش 1-4 مورد بررسی قرار گرفت.

۴-۷) متوقف کردن و خاتمه دادن برنامه

خلاصه: گزینه های **Halt** و **Terminate** موجود در دیباگر برای ایجاد توقف بطور موقت یا دائم در برنامه بکار می روند. گاهی اوقات اجرا برنامه بسیار طول می کشد. و ما متعجب از این که همه چیز درست هست. شاید برنامه شما در یک حلقه نامتناهی گرفتار آمده و به همین دلیل اجرا برنامه به طول می انجامد. خب ما این مورد را بررسی می کنیم. متد **longloop()** را از کلاس **Demo** انتخاب نمایید. حالا ما قصد داریم ببینیم که برنامه به کجا منتقل می شود. پنجره دیباگر را نگاه کنید اگر هنوز بر روی صفحه نمایش ظاهر نشده است. حالا دکمه **Halt** را فشار دهید. اجرا برنامه اگر ما نقطه **break point** تعیین کرده باشیم متوقف می شود. شما حالا می توانید مرحله به مرحله پیش بروید و از متغیرها رو مشاهده کنید که همه مقدار درستی دارند. این عمل فقط نیاز به مقداری زمان برای تکمیل شدن دارد. شما می توانید به دفعات زیاد از دکمه های **Halt** و **Continue** استفاده کنید تا بتوانید سریعتر پیش بروید. اگر شما قصد ندارید که ادامه دهید (به عنوان مثال جای واقعی حلقه بی نهایت را پیدا کرده اید) می توانید دکمه **terminate** را فشار دهید تا اجرای برنامه متوقف شود. شما نمی توانید از دکمه **Terminate** مکررا استفاده نماید.

فصل ۸ - ایجاد یک برنامه کاربردی خود اتکا

خلاصه : برای ایجاد یک برنامه خود اتکا از منوی **project** گزینه **create jar file** را انتخاب نمایید.

BlueJ می تواند فایل های **Jar** قابل اجرا تولید نماید. فایل های **Jar** اجرایی ، در سیستم های نظیر ویندوز و مکینتاش با

دو بار کلیک کردن بر روی فایل اجرا می شوند. و در سیستم های نظیر **Dos** و **Unix** با تایپ دستور زیر :

Java -jar <file name>.jar

ما این کار را برای پروژه **people** که در پوشه **examples** قرار دارد. انجام می دهیم. پروژه را باز کنید. مطمئن شوید که پروژه

کامپایل شده و بعد از منوی **Project** گزینه **create jar file** را انتخاب نمایید.

پنجره ای باز می شود. که به شما اجازه می دهد تا کلاس **main** را انتخاب نمایید (عکس ۱۸). این کلاس باید شامل یک

متد **main** (که امضا آن بدین شکل باشد : **public static void main(String[] args)**) باشد.



عکس ۱۸ : پنجره "create jar file"

در مثال ما انتخاب کلاس **main** ساده می باشد. زیرا ما تنها یک کلاس داریم. بنابراین از منوی بازشونده **Hello** را انتخاب نمایید.

معمولا شما سورس برنامه را به همراه فایل اجرایی قرار نمی دهید. ولی اگر قصد دارید که برنامه تان شامل سورس برنامه هم باشد . می توانید گزینه **include source** را تیک بزنید. (به عنوان مثال شما می توانید کل پروژه تان را با استفاده از فرمت **jar** در قالب یک فایل بوسیله ایمیل برای دیگران ارسال نمایید) .

اگر شما **BlueJ** را به گونه ای تنظیم کرده اید. که

از **user library** (برای اینکار مسیر **Tools / performance / libraries** را طی کنید) استفاده کند. در وسط این پنجره شما ناحیه ای را با این عنوان مشاهده خواهید کرد : **"Include user libraries"** . (اگر شما از کتابخانه ها استفاده

نکرده اید. این ناحیه نمایش داده نمی شود) شما می توانید انتخاب کنید کتابخانه های مختلفی را که پروژه شما از آن ها استفاده می کند. Continue را فشار دهید تا به مرحله بعد بروید. در این بخش یک پنجره انتخاب کننده فایل (file chooser) نمایش داده می شود. که شما با استفاده از آن نام فایل Jar را تایپ کرده و دکمه Create را فشار دهید. برای این مثال شما تایپ کنید: Hello و دکمه create را فشار دهید. اگر شما به فایل تان کتابخانه ای را اضافه نکرده باشید. فایل hello.jar ایجاد می شود. و اگر برنامه شما شامل کتابخانه ای باشد. پوشه ای با نام hello ایجاد شده که فایل hello.jar در آن قرار دارد. همچنین این پوشه شامل تمامی کتابخانه های مورد نیاز می باشد. فایل Jar شما کتابخانه های مورد نیاز خود را در پوشه ای همنام با خودش می یابد. بنابراین همواره فایل Jar را یا این پوشه جابجا کنید. تا فایل Jar بتواند کتابخانه های مورد نیاز را پیدا نماید. اگر برنامه شما شامل یک رابط گرافیکی کاربر (GUI) می باشد. با دو بار کلیک بر روی فایل Jar می توانید آنرا اجرا کنید. و اگر مانند مثال ما فایل Jar تنها شامل یک I/O می باشد. شما با استفاده از Text Terminal و یا پنجره داس می توانید آنرا اجرا کنید. برای اینکار مراحل زیر را طی کنید:

در پنجره داس یا Text Terminal به محلی که فایل jar قرار دارد رفته و دستور زیر را تایپ کنید:

```
Java -jar hello.jar
```

بدین گونه فایل شما اجرا می شود.

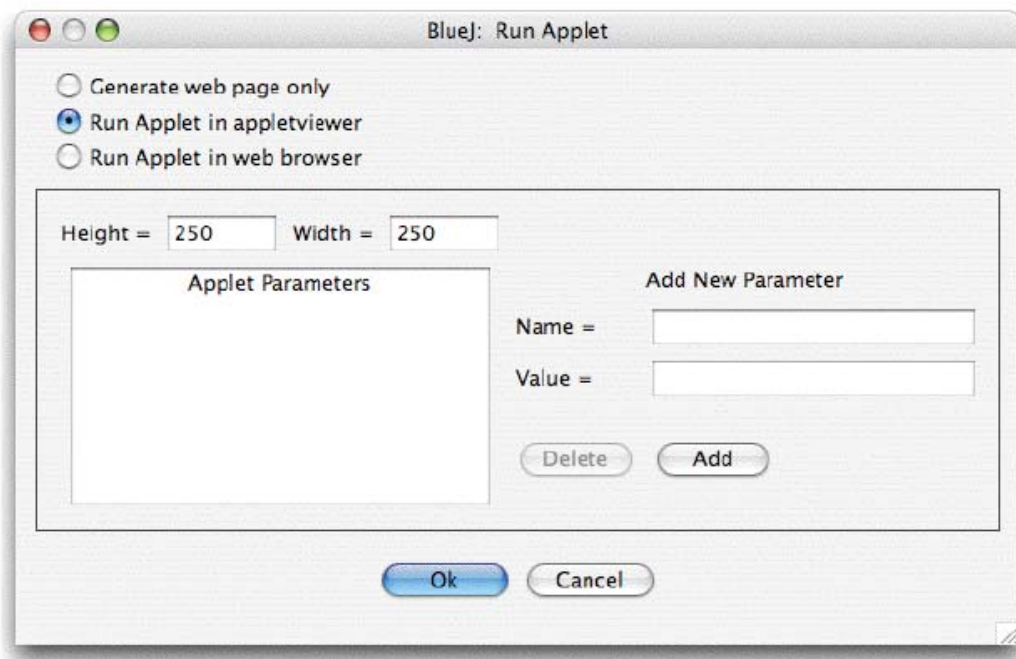
۹-۱) اجرا کردن اپلت ها

خلاصه: برای اجرا کردن اپلت ها از منوی بازشونده **applet** گزینه **Run Applet** را انتخاب نمایید.

BlueJ به شما اجازه می دهد که اپلت ها را مانند اپلیکیشن ها ایجاد و اجرا کنید. به همراه **BlueJ** و در پوشه **Examples** یک نمونه اپلت وجود دارد. ما در ابتدا قصد داریم تا اپلت را اجرا نماییم. برای اینکار پروژه **Appletdemo** را از پوشه **examples** انتخاب نمایید.

شما خواهید دید که این پروژه تنها شامل یک کلاس با نام **CaseConvertor** میباشد. این کلاس بوسیله تگ **<<applet>>** مشخص شده است. بعد از کامپایل پروژه، گزینه **Run Applet** را از منوی باز شونده کلاس انتخاب نمایید.

پنجره ای باز می شود که به شما یک سری گزینه ارائه می دهد (عکس ۱۹).



عکس ۱۹: پنجره Run Applet

شما در این پنجره می بینید که می توانید انتخاب کنید که اپلت در یک مرورگر نمایش داده شود و یا اینکه در **Applet viewer** و یا اینکه تنها **webpage** مربوطه تولید شده و اپلت اجرا نشود. تنظیمات پیش فرض را دست نزنید و دکمه **Ok** را فشار دهید.

بعد از چند ثانیه **Applet viewer** نمایش داده شده و اپلت شما را نمایش می دهد.

Applet Viewer به همراه **J2SE** بر روی سیستم شما نصب شده و همواره این تضمین وجود دارد که نسخه کامپایلر جاوا ای شما و **Applet viewer** یکسان باشد. در مورد مرورگر وب شما امکان دارد. که مرورگر از نسخه دیگری از جاوا استفاده کند و این مورد باعث بروز یک سری مشکل شود. البته در حال حاضر اکثر مرورگر ها بخوبی با جاوا کار می کنند.

در سیستم عامل ویندوز و مکینتاش، **BlueJ** از مرورگر پیش فرض سیستم عامل استفاده می کند. اما در سیستم های **Unix**، مرورگر در تنظیمات **BlueJ** مشخص شده است.

۹-۲) ایجاد کردن اپلت

خلاصه: برای ایجاد اپلت بر روی دکمه **New Class** کلیک کرده و **Applet** را از نوع کلاس انتخاب نمایید. بعد از این که مشاهده کردید چگونه می توان یک اپلت را اجرا کرد. حالا قصد داریم تا خودمان یک اپلت ایجاد نماییم. یک کلاس جدید با فرمت **Applet** ایجاد کنید. انرا کامپایل کنید. و در نهایت اجرا کنید. این کار انجام شد !!! حالا می توانید کدهای اپلت را در ادیتور ببینید. و مشاهده کنید که شامل تمامی کدهای مورد نیاز برای اپلت می باشد. به عنوان مثال متد **paint** .

۳-۹) آزمایش کردن اپلت

در بسیاری از حالت ها ایجاد شی اپلت بر روی میز کار اشیا بسیار مفید خواهد بود. شما می توانید از منوی بازشونده اپلت سازنده اپلت را مشاهده نمایید. از میز کار اشیا شما نمی توانید یک اپلت را به طور کامل اجرا کنید. بلکه بعضی از متدهای آن را می توانید فراخوانی نمایید. این موضوع برای تست کردن یک متد که شما در بین کدهای اپلت نوشته اید می تواند مفید باشد. اگر شما برای اپلت تان نقاط **break point** تعیین کرده اید. در حالتی که اپلت در **Applet viewer** و یا مرورگر وب اجرا می شود. هیچ گونه تاثیری ندارد. زیرا آنها از ماشین های مجازی مربوط به خود بهره می برند. و نمی توانند معنای نقاط **break point** را که نرم افزار **BlueJ** قرار داده درک کنند.

اگر شما قصد دارید تا از **Break point** ها و همچنین کامپایل مرحله به مرحله در اپلت تان استفاده کنید. شما می توانید از کلاس **AppletWindow** که بوسیله مایکل تریکویوف نوشته شده استفاده نمایید. این کلاس برای شما فریمی را بوجود می آورد. که با استفاده از آن می توانید اپلت ها را مستقیما در درون **BlueJ** اجرا نماید. به گونه ای که متدهای عادی مربوط به رفع خطا کار کنند. شما این کلاس و یک نسخه نمونه از آن را می توانید در بخش **Resource** سایت **BlueJ** بیابید.

فصل ۱۰ - اعمال دیگر

۱-۱۰) باز کردن پکیج های غیر BlueJ در BlueJ

خلاصه : برای باز کردن پکیج های غیر BlueJ از منوی Project گزینه Open Non-BlueJ را انتخاب نمایید.
BlueJ به شما اجازه می دهد تا پکیج های را که در خارج از محیط BlueJ ایجاد شده در درون BlueJ مشاهده نمایید.
برای انجام اینکار از منوی Project گزینه Open Non BlueJ را انتخاب کرده . سیستم از شما مکان و پوشه ای را که قصد دارید آنرا باز کنید. سوال خواهد کرد.

۲-۱۰) اضافه کردن کلاس های موجود به پروژه تان

خلاصه : کلاس ها را می توانید از جایی خارج از پروژه ، به پروژه اضافه کنید . برای اینکار از گزینه Add Class From file بهره خواهیم جست.

گاهی اوقات شما نیاز دارید تا از کلاسی استفاده کنید که در خارج از محیط BlueJ قرار دارد. به عنوان مثال استاد شما به شما کلاس جاوای داده تا از آن در پروژه تان استفاده کنید. شما می توانید به سادگی با انتخاب گزینه Add Class Form file از منوی Edit برنامه ، کلاسی را که در خارج از پروژه تان قرار دارد. به پروژه اضافه نمایید. شما با این روش فایل های سورس جاوا (یعنی فایل های که به java. ختم شده اند) را به پروژه تان وارد نمایید. هنگامی که شما کلاس مورد نظر را به پروژه تان اضافه می کنید. یک نسخه از آن در پوشه پروژه کپی می شود. این عمل دقیقا شبیه به این است که شما یک کلاس ایجاد کرده و تمام کدهای آنرا خودتان تولید نمایید.
روش دیگر برای اضافه کردن یک کلاس به پروژه ، بدین صورت می باشد که شما فایل سورس جاوا را در پوشه پروژه کپی کنید. هنگامی که دوباره پروژه را بوسیله BlueJ باز کنید. خواهید دید که BlueJ کلاس مورد نظر را در نمایش ترسیمی نمایش خواهد داد.

۳-۱۰) فراخوانی متدهای main و static :

خلاصه : متدهای static را می توان از منوی بازشونده کلاس فراخوانی کرد.
پروژه hello را از پوشه examples باز کنید. تنها کلاس این پروژه (Hello Class) بوسیله یک متد main استاندارد تعریف شده است. بر روی کلاس کلیک راست کرده ، منوی کلاس تنها شامل سازنده کلاس نیست . بلکه متد استاتیک main هم نمایش داده می شود. شما می توانید متد mian را از این منو فراخوانی نمایید. تمام متدهای استاتیک را می توان شبیه به همین مورد فراخوانی نمود.
متد استاندارد main شامل آراییه ای از رشته ها به عنوان پارامتر می باشد. شما می توانید آراییه ای از رشته ها را بر اساس قواعد جاوا برای این متد ارسال کنید. به عنوان مثال :

```
{"one", "two", "three"}
```

در ضمن باید براکت ها را هم به عنوان پارامتر ارسال کنید.

توجه: در جاوا استاندارد شما نمی توانید از آرایه های ثابت به عنوان آرگومان برای فراخوانی متدها استفاده کنید. شما از آن ها فقط برای دادن مقدار اولیه می توانید استفاده کنید. در BlueJ ما به شما اجازه می دهیم که شما برای تعامل و فراخوانی متد Main از آرایه های ثابت به عنوان پارامتر استفاده نمایید.

۴-۱۰ تولید اسناد

خلاصه : برای تولید سند برای پروژه تان ، از منوی Tools گزینه Project Documentation را انتخاب کنید. شما بوسیله BlueJ می توانید برای پروژه تان در فرمت استاندارد javadoc سند تولید نمایید. برای انجام این کار از منوی Tools گزینه Project Documentation را انتخاب نمایید.

با این کار برنامه برای تمام کلاس های موجود در پروژه تان با استفاده کلاس های سورس شان سند تولید می کند و مرورگر وب را برای نمایش آنها باز می نماید.

شما همچنین می توانید سند را برای یک کلاس تولید کرده و ببینید. برای انجام این کار به محیط کد نویسی کلاس مربوطه بروید. و حالت Implementation را به Interface تغییر دهید. در این حالت شما سند javadoc کلاس مربوطه را خواهید دید.

۵-۱۰ کار کردن با کتابخانه ها

خلاصه : شما با انتخاب گزینه Java Class Libraries از Help می توانید کلاس های استاندارد جاوا را مشاهده کنید. همانطور که می دانید ؛ هنگامی که شما با جاوا یک برنامه می نویسید. شما به کتابخانه استاندارد جاوا مراجعه می کنید. شما برای دیدن اسناد مربوط به این کتابخانه ها می توانید از منوی Help گزینه Java Class Libraries را انتخاب کنید. (البته شما باید Online باشید)

البته اگر اسناد جاوا را به طور محلی نصب کرده اید در حالت offline هم می توانید این کار را انجام دهید. برای دیدن جزئیات این کار به سایت BlueJ مراجعه نمایید.

۶-۱۰ ایجاد اشیا از کلاس های کتابخانه ای

خلاصه : برای ایجاد اشیا از کلاس های کتابخانه ای از منوی Tools گزینه Use Library Class را انتخاب نمایید.

BlueJ همچنین توابعی را ارائه می دهد تا شما بتوانید از کلاس های که جز پروژه تان نیست ولی در کتابخانه موجود است اشیای ایجاد کنید. برای مثال شما می توانید اشیایی از کلاس های String و یا ArrayList ایجاد کنید. این مورد برای تجربه کردن کار کردن اشیا این کلاس ها می تواند خیلی مفید باشد.

شما با انتخاب Use Library Class از منوی Tools می توانید از کتابخانه یک شی ایجاد کنید. پنجره ای باز می شود که به شما اجازه می دهد تا نام کامل کلاس را وارد کنید، مانند java.lang.String (باید توجه کنید که نام کامل کلاس باید شامل پکیج های که کلاس جز آن است باشد)

محل ورود متن (شامل یک منوی بازشونده است که شامل نام آخرین کلاس های است که شما استفاده کرده اید با انتخاب

یکی از این کلاس ها و فشار دادن دکمه Enter ، تمام سازنده ها و متدهای static کلاس مربوطه لیست می شود. هر کدام از این سازنده ها و یا متدهای static را با انتخاب کردن می توان مورد استفاده قرار داد.

فصل ۱۱ -- خلاصه ها

شروع کردن

۱. برای باز کردن پروژه از منوی Project گزینه open را انتخاب کنید.
۲. برای ایجا یک شی ، سازنده (constructor) را از منوی بازشونده Class (popup) انتخاب کنید.
۳. برای اجرایی کردن یک متد ، آنرا از منوی بازشونده شی (object) انتخاب کنید.
۴. برای ویرایش کردن سورس یک کلاس بر روی آیکون کلاس دو بار کلیک کنید.
۵. برای کامپایل یک کلاس ، دکمه compile ادیتور را فشار دهید. و برای کامپایل یک پروژه دکمه compile پنجره پروژه را فشار دهید.
۶. اگر در هنگام کامپایل کردن با پیام خطایی مواجه شدید. با فشار دادن علامت سوال (؟) در محیط کدنویسی اطلاعات بیشتری در مورد آن خطا بدست آورید.

انجام اعمال بیشتر

۷. ناظر اشیا به شما اجازه می دهد با چک کردن موقعیت درونی اشیا یک سری از رفع خطاهای ساده را انجام دهید.
۸. با کلیک کردن بر روی آیکون اشیا می توان پارامترهای را به متدهای آنها ارسال نمود.

ایجاد پروژه جدید

۹. برای ایجاد پروژه جدید از منوی Project گزینه New.... را انتخاب نمایید.
۱۰. برای ایجاد کلاس جدید دکمه New class را فشار دهید. نام مشخصی برای کلاس قرار دهید.
۱۱. برای ایجاد یک پیکان (Arrow) . دکمه Arrow را فشار دهید. و آنرا بر روی صفحه طراحی Drag کنید. یا اینکه می توانید مستقیما در محیط کدنویسی ، کد مربوطه را وارد نمایید.
۱۲. برای حذف کلاس و یا یک پیکان ، از منوی بازشونده گزینه remove را انتخاب نمایید.

استفاده از محیط کدنویسی

۱۳. برای شروع کار با محیط کدنویسی ، از منوی view گزینه Show Code Pad را انتخاب نمایید.
۱۴. برای امتحان کردن دستورات جاوا ، آنها را در محیط کدنویسی تایپ نمایید.
۱۵. برای انتقال اشیا از محیط کدنویسی به شاخه اشیا ، آیکون کوچک اشیا را درگ کنید.

۱۶. برای دیدن اشیا در محیط کدنویسی ، بر روی آیکون کوچک اشیا دوبار کلیک کنید.
۱۷. دستوراتی که در محیط کدنویسی تایپ می نمایید. اجرایی هستند.
۱۸. برای وارد کردن دستورات چند خطی در پایان خط **enter+shift** را فشار دهید.
۱۹. متغیرهای محلی می توانند در دستورات تک خطی یا چند خطی استفاده شوند. نام های اشیا در فیلدهای نمونه اشیا ذخیره می شوند.
۲۰. با استفاده از نشانگرهای بالا و پایین صفحه کلید برای دیدن تاریخچه ورودی ها استفاده نمایید.

رفع خطا

۲۱. برای ایجاد Breakpoint در منطقه breakpoint در سمت چپ محیط کدنویسی کلیک نمایید.
۲۲. برای دیدن کامپایل خط به خط برنامه از گزینه های Step و Step into در دیباگر برنامه استفاده نمایید.
۲۳. نظارت کردن بر متغیرها ساده است – تمام آنها بصورت اتوماتیک در دیباگر نمایش داده می شوند.
۲۴. ایجاد توقف یا خاتمه دادن به برنامه را می توانید با گزینه های Halt و Terminate دیباگر انجام دهید.

ایجاد برنامه های خود اتکا (خود اجرا)

۲۵. برای ایجاد برنامه های خود اجرا از منوی project گزینه create jar file... را انتخاب نمایید.

ایجاد اپلت ها

۲۶. برای اجرا اپلت ها از منوی بازشونده اپلت گزینه Run Applet را انتخاب نمایید.
۲۷. برای ایجاد اپلت ها دکمه New class را فشار داده و گزینه Applet را از انواع کلاس انتخاب نمایید.

اعمال دیگر

۲۸. پکیج های غیر blueJ را می توان با استفاده از منوی Project و انتخاب گزینه Open Non BlueJ باز کرد.
۲۹. کلاس های که خارج از محیط BlueJ هستند را می توانید با استفاده از منوی Edit و انتخاب گزینه Add Class from file..... انتخاب نمایید.
۳۰. متدهای static کلاس را می توانید با استفاده از منوی بازشونده کلاس فراخوانی نمایید.
۳۱. برای تولید سند برای پروژه تان از منوی Tools گزینه Project Documentation را انتخاب نمایید.
۳۲. برای دیدن کلاس های استاندارد جاوا از منوی Help گزینه java standard Libraries را انتخاب نمایید.
۳۳. برای ایجاد اشیا از کلاس های کتابخانه ای از منوی Tools گزینه Use Class Library را انتخاب نمایید.