Click Here

```
David J. Barnes & Michael Kölling Objects First with Java A Practical Introduction using Blue J Seventh Edition): 978-0-138-087173 ISBN (Global Edition): 978-1-924-63599 These are the instructions for the generic Blue J installer (the installer (the installer (blue J installer (blu
operating systems as well as Mac OS X and Windows. However, it is generally preferable to use the "native" installer when available. The generic installer requires that a Java JDK be installed separately. There are two possibilites depending on the version of BlueJ: For BlueJ 5.4.0 and Vindows. However, it is generally preferable to use the "native" installer when available. The generic installer when available.
available for 64-bit operating systems. You can download the OpenJDK from Adoptium: choose version "21 - LTS" and then your operating system from Gluon. This is a zip which you should unzip locally. The BlueJ installer will ask you for the path
of the JDK and unzipped JavaFX. For Blue JavaFX. For Blue
operating system from Gluon. This is a zip which you should unzipped JavaFX. For Blue and unzipped JavaFX. For Blue and unzipped JavaFX. For Blue and unzipped JavaFX. For Blue are only available for 64-bit operating systems. You can download the Open DK 11 and Open DK 11 are required, which are only available for 64-bit operating systems.
you must unzip locally. You also need OpenJFX 11 -- download the SDK download for your operating system from Gluon. Again, this is a zip which you should unzip locally. The BlueJ installer will ask you for the path of the unzipped directories. For BlueJ 4.0.0 to 4.1.4 a Java 8 JDK is required, and we recommend installing the latest JDK 8 update -- you
can download the Oracle JDK from the Oracle yebsite. Make sure to get the JDK (not the JRE) and to download the correct version. Installation Procedure Download the installation by typing some commands
into this application. (For Windows users, see these instructions for how to start the Command Prompt). Run the following command: java -jar ... replacing with the correct path to the java executable. For example:
"C:\Program Files\Java\jdk1.8.0 144\bin\java.exe" -jar "C:\Users\Sarah\Downloads\bluej-411.jar" (this example is for Windows; other operating system path to the JDK, and for BlueJ 4.2.0 the path to OpenJFX, (make sure it is
correct!) and where you want BlueJ to be installed. It is usually best to installed BlueJ: cd ... replacing with the path you
chose to install Blue to. Start the Blue batch file: blue
Practical Introduction using BlueJ Chapter 3 Objects Chapter 5 Functional Processing of Collections Chapter 6 More-Sophisticated Behavior Chapter 7 Fixed-Size Collections—Arrays Chapter 8 Designing Classes
Chapter 9 Well-Behaved Objects Chapter 12 Improving Structure with Inheritance Chapter 13 More About Inheritance Chapter 14 Further Abstraction Techniques Chapter 15 Data-Oriented Classes Chapter 15 Data-Oriented Classes Chapter 16 A Brief History of Java Chapter 17 Building Graphical User Interfaces Chapter 18
Handling Errors Chapter 19 Designing Applications Chapter 20 A Case Study Back to book description Back to book descripti
Raspberry Pi BlueJ's functionality can be enhanced using a variety of available extensions. The FAQ — frequently asked questions. Problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above: Support Request Form — ask for help if you have problems not found in either of the above problems not found in either of t
with BlueJ. A number of journal and conference papers about BlueJ and computer science education in general: [PDF] Kouznetsova, S., Using BlueJ and Blackjack to teach object.
oriented design concepts in CS1, Journal of Computing Sciences in Colleges, Volume 22 Issue 4, April 2007, This paper describes a course that uses an implementation of the card game 'BlackJack' with BlueJ as an introduction to OO programming (presented at CCSC: Central Plains 2007 conference, April 2007) [PDF] Van Haaster, K. and Hagan, D.,
Teaching and Learning with BlueJ: an Evaluation of a Pedagogical Tool, Information Science + Information Technology Education, Special issue on
Learning and Teaching Object Technology, Vol 13, No 4, Dec 2003. This is a general BlueJ overview paper. Something we should have written long ago, but didn't. It explains some of the basic thoughts behind BlueJ, Proceedings of
the 8th conference on Information Technology in Computer Science Education (ITiCSE 2003), Thessaloniki, 2003. This paper describes the idea behind JUnit integration in BlueJ, its design and the effect for the user. [PDF] Kölling, M. and Rosenberg, J., Guidelines for Teaching Object Orientation with Java, Proceedings of the 6th conference on
Information Technology in Computer Science Education (ITiCSE 2001), Canterbury, 2001. A few hints about how to design assignments for introducing Java programming (with our without BlueJ). These are some general thoughts - this is then presented in more concrete terms in The BlueJ system and its pedagogy above. [PDF] Kölling, M., Teaching
Object Orientation with the Blue Environment, Journal of Object-Oriented Programming, Vol. 12 No. 2, 14-23, 1999. [PDF] Kölling, M., The Blue Language for Teaching, PhD Thesis, Basser Department of Computer Science, University of Sydney, 1999. [PDF] Kölling, M., The Blue Language for Teaching, PhD Thesis, Basser Department of Computer Science, University of Sydney, 1999. [PDF] Kölling, M., The Blue Language for Teaching, PhD Thesis, Basser Department of Computer Science, University of Sydney, 1999. [PDF] Kölling, M., The Blue Language for Teaching, PhD Thesis, Basser Department of Computer Science, University of Sydney, 1999. [PDF] Kölling, M., The Blue Language for Teaching, PhD Thesis, Basser Department of Computer Science, University of Sydney, 1999. [PDF] Kölling, M., The Blue Language, Journal of Object-Oriented Environment, Inc., 1999. [PDF] Kölling, M., The Blue Language, Inc.
Oriented Programming, Vol. 12 No. 1, 10-17, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Part 1: Languages, Journal of Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Part 1: Languages, Journal of Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Part 1: Languages, Journal of Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 12 No. 1, 10-17, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 12 No. 1, 10-17, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 6-12, 1999. [PDF] Kölling, M., The Problem of Teaching Object-Oriented Programming, Vol. 11 No. 9, 1999. [PDF] Kölling, M., The Problem Object-Ori
Programming, Vol. 11 No. 8, 8-15, 1999. [PDF] Kölling, M. and Rosenberg, J., Support for Object-Oriented Languages and Systems (TOOLS) 28, IEEE, Melbourne, 204-215, 1998. [PDF] Rosenberg, J. and Kölling, M., I/O Considered Harmful (At least for the first few weeks), in Proceedings of the Second
Australasian Conference on Computer Science Education, ACM, Melbourne, 216-223, July 1997. [PDF] Rosenberg, J. and Kölling, M., Testing Object-Oriented Programs: Making it Simple, in Proceedings of 28th SIGCSE Technical Symposium on Computer Science Education, ACM, San Jose, Calif., 77-81, February 1997. [PDF] Kölling, M. and
Rosenberg, J., An Object-Oriented Program Development Environment for the First Programming Course, in Proceedings of 27th SIGCSE Technical Symposium on Computer Science Education, ACM, Philadelphia, Pennsylvania, 83-87, March 1996. [PDF] Kölling, M. and Rosenberg, J., Blue - A Language for Teaching Object-Oriented Programming, in
Proceedings of 27th SIGCSE Technical Symposium on Computer Science Education, ACM, Philadelphia, Pennsylvania, 190-194, March 1996. [PDF] Kölling, M., B. Koch and Rosenberg, J., Requirements for a First Year Object-Oriented Teaching Language, in ACM SIGCSE Bulletin, ACM, Nashville, 173-177, March 1995. Blue I now supports Git
repositories as part of its teamwork features. This tutorial briefly explains how to use the new Git support on Bluej. Contents Introduction to version control and Git It is important to learn some basic Git terminology: Working copy: The place where your Bluej shared project is located. Whenever you make any changes to your files in your shared
BlueJ project, you are making those changes in the working copy. Local repository is stored in the same directory as the BlueJ project itself. It is a subdirectory as the BlueJ project itself. It is a subdirectory called '.git' that contains all the files needed in order to perform the version control operations, etc.). You should not touch this as the BlueJ project itself. It is a subdirectory called '.git' that contains all the files needed in order to perform the version control operations, etc.).
directory. Remote repository: A remote Git repository is used to keep track of your shared BlueJ project, but resides in another computer. It is usually used to share code with other developers: you can push your changes from the remote repository into your working copy. Creating a Git repository
There are several web-based Git repository hosting service. They can be a good option if you want to very quickly and easily create repositories to host your BlueJ projects. We talk about some of the most popular a web-based Git repository hosting service. It
provides a web-based graphical interface where you can see your repository, its files, commits, etc. It also provides forums where you can discuss features in your projects, and a wiki engine where you can discuss features in your projects, and a wiki engine where you can discuss features in your projects, and a wiki engine where you can discuss features in your projects, and a wiki engine where you can discuss features in your projects, and a wiki engine where you can discuss features in your projects. In order to use GitHub, you should create a completely empty repository. You can
find a tutorial on how to create your GitHub repository here. Important: in step 5, UNTICK "Select Initialize this repository with a README", so that you get an empty repository using the Git command line commands (Figure 1), but this should be
ignored: BlueJ will will perform them automatically when checking out a project. Make note of the repository address (below the phrase: "Quick setup - if you've done this kind of thing before"): you will need to use it together with your GitHub username and password to share a project into or checkout a project from this repository (see the
instructions in the following sections). Figure 1: After creating a new repository in GitHub: quick setup screen On GitHub we recommend you use the https URI, since BlueJ uses username and password authentication. Bitbucket Bi
can create free private repositories limited to 5 users. You can find a tutorial on how to create a Git repository in Bitbucket here. Important: change the "Include a README?" setting to No to get an empty repository and has integrated code reviews, issue
tracking, etc. Differently from GitHub and Bitbucket, it can be installed in your local network. Given you have GitLab installed in your network, You can find a tutorial on how to create a Git repository set, you may want to share your BlueJ project with other users. In
order to do so, open the existing project you want to share. Select: Tools -> Team -> Share this project menu option (Figure 2). You should see a dialog similar to the one in Figure 3. Figure 2: Sharing a project. Select: Tools -> Team -> Share this project menu option (Figure 2). You should see a dialog similar to the one in Figure 3. Figure 3
Your e-mail: The e-mail associated to the changes you make to the repository. User: The username used to access the remote repository. This is usually your login in the remote repository. URI: A URI containing the address of your Git repository (ex: .
Figure 3: Settings for sharing/checking out a project with Git. Tip: It is good practice to click on the "Check connection" button: this button tests if BlueJ can connect to the remote repository and if the URI does contain a valid Git repository and if the URI does contain a valid Git repository in BlueJ can connect to the remote repository and if the URI does contain a valid Git repository.
repository, you need to create a local clone of the remote Git repository (this operation should be done only once). This local clone of the project can later be open and used just like a normal BlueJ project. To create the local clone of the project can later be open and used just like a normal BlueJ project. To create the local clone of the project can later be open and used just like a normal BlueJ project. To create the local clone of the project can later be open and used just like a normal BlueJ project.
Tools->Team->Checkout project... Then a dialog similar to the one in Figure 3 will appear. Note: The Check out command in BlueJ is acutally performing a 'clone' operation to create a local copy of the repository, with the main branch checked out into the working copy. You should select "Git" as your server type. This will allow you to enter: Your
name: The name associated to the changes you make to the repository. Your e-mail: The e-mail associated to the repository. User: The username used to access the remote machine, and it is usually case sensitive. Password: The password associated to the username
Repository URI: A URI containing the address of your Git repository containing the BlueJ can connect to the remote repository and if the URI does contain a valid Git repository. After clicking on "OK", you should now select where you want BlueJ to checkout the project (ex: . Tip: Click on the "Check connection" button: this button tests if BlueJ can connect to the remote repository and if the URI does contain a valid Git repository.
in your machine. You should choose a name for your project, the project, therefore, it is likely that either you or other users made changes to the project, therefore, it is important to know if you have changes that are ready to commit to your local repository, or changes
ready put in the remote repository (push), or if you need to update your local project because someone made a change in the remote repository. This information is presented on the status dialog, which can be displayed by pressing the "Status" button on BlueJ main screen (Figure 4: BlueJ main screen. You can see the status dialogue below
Figure 5: BlueJ Status dialogue with local changes. The Figures 5 and 6 have three columns: File name: The files that are being tracked by the version control. These are the project java files, the Project's README.TXT file, any subfolders and the Diagram layouts (a Bluej special file that helps drawing the class diagrams). Local status: The status of
the files in the working copy relative to the local repository. Remote status: The status of the listed files relative to remote repository: in Figure 9, we can see that a push action would create "NewClass.java", modifty "README.TXT" and remove "Class2.java". A file can have one of the following Local status: Blank: The file is up to date in the local
repository and no changes should take place. New: The file is new in the working copy. A commit would make Git track the file and add it to the local repository, but has been modified in the working copy. A commit would save those changes to the local repository, but has been modified in the working copy. A commit would save those changes to the local repository, but has been modified in the working copy. A commit would save those changes to the local repository, but has been modified in the working copy.
repository, but it was deleted. A commit would remove the file from the repository. Needs pull: This status can only appear in the remote repository, and it is always in conjuction to one of the previous status, indicating which effect a pull would make in the remote repository. Needs update: This status can only appear in the remote repository, and
indicates that your local file is older in comparison to the one in the remote repository; you need to perform an update. A file can have one of the following Remote status: Blank: The file is up to date in the remote repository, and it is always in
conjuction to one of the previous status, indicating which effect a pull would make in the remote repository, and indicates that your local file is older in comparison to the one in the remote repository; you need to perform an update. Figure 6: BlueJ Status dialogue with changes to be
pushed. Committing and pushing changes Commit As you may share those changes into your local repository. In Figure 7, we show the commit/push window. In the window we can see a list of files ready to
commit: These files have modified in some way since the last time an update was made: you may have delete some of them (Class2.java); or created new ones (NewClass.java). You will need to write a comment in your commit: what are the new changes that you made in this commit? this is important as a way of describing what these changes do. This
comment will be written in the repository log and are a useful way of quickly and clearly communicating your changes to other users. The commit is done by pressing the Commit button. You don't need to be connected in order to make commits: Git is a distributed version control system, which means that you have a fully functional local repository
inside your Blue project, and when you commit you are actually putting your changes to this local repository. Figure 7: Commiting your changes to the pushed to the remote repository, where they can be shared by the other users. You can see the list of files
ready to be pushed in Figure 5. Figure 8: Pushing changes into the remote repository. You need to be able to connect to the remote repository when you push. BlueJ may ask for the password to connect to the remote repository when you push. BlueJ may ask for the password to connect to the remote repository.
affected by each commit and the user who made the commit. You can access the repository log by opening the menus Tools->Team->Project History. An example of an entry can be seen in Figure 9. Figure 9. Figure 9. Figure 9. Figure 9. Figure 9. The project history entry. In big projects, or projects with many participants you may have a quite long history. For that purpose, we added two
filters at the end of the window: "Show file" where we display only the commits that contains the selected file, and "Show user", where you can see the only the commits of the selected user. Update window (Figure 10), there you will see the list of files affected by this
update. To update, you should press the "update" button. Figure 10: The Update project window. Support Please see support page. Objects First with Java A Practical Introduction using BlueJ This document describes the goals, contents of this
description: Java BlueJ Real objects first An iterative approach Concept sequence This book is an introduction to object-oriented and programming for beginners. The main focus of the book is general object-oriented and programming concepts from a software engineering per-
spective. While the first chapters are written for students with no programmers as well. In particular, programmers as well. In particular, programmers as well. In particular, programmers as well also be able to benefit from the
book. We use two tools throughout the book to enable the concepts introduced to be put into practice: the Java programming language and the Java programming language itself provides a clean implementation of most of the
important object-oriented concepts, and serves well as an introductory teaching language. Its popularity ensures an immense pool of support resources. In any subject area, having a variety of sources of information available is very helpful, for teach-ers and students alike. For Java in particular, countless books, tutorials, exercises, compilers, en-
vironments, and quizzes already exist, in many different kinds and styles. Many of them are online and many are available free of charge. The huge amount of high quality support material already exist, in many different kinds and styles. Many of them are online and many are available free of charge. The huge amount of high quality support material makes Java an excellent choice as an introduction to object-oriented programming. With so much Java material already exist, in many different kinds and styles.
London, UK, explicitly as an environment for teaching introductory object-oriented programming. It is better suited to introduction. From
then on, instruction can concentrate on the important concepts at hand—object orientation and Java—and no time needs to be wasted talking about environments, file systems, class paths, or DLL conflicts. The environments supports important teaching tools not available in other environments. One of them is visualization of class structure. BlueJ
automatically displays a UML-like diagram rep-resenting the classes and relationships in a project. Visualizing these important concepts is a great help to both teachers and students. It is hard to grasp the concept of an object when all you ever see on the screen is lines of code! The diagram notation is a simple subset of UML, tailored to the needs of
beginning students. This makes it easy to understand, but also allows migration to full UML in later courses. One of the most important strengths of the BlueJ environment is the user's ability to directly create objects of any class, and then to interact with their methods. This creates the opportunity for direct experimentation with objects, with little
teamwork using a ver-sion control system, such as GIT, once the students are ready. Several of these features are unique to the BlueJ environment. It is not a cut-down, simplified version of Java for teaching. It runs on top of the OpenJDK Java Development Kit, and makes use of the standard compiler and virtual
machine. This ensures that it always conforms to the official and most up-to-date Java specification. The authors of this book have many years of teaching experience with the BlueJ environment (and many more years without it before that). We both have experienced how the use of BlueJ has in-creased the involvement, understanding, and activity of
students in our courses. One of the au-thors is also the development lead of the BlueJ system. Real objects first One of the reasons for choosing BlueJ was that it allows an approach where teachers for some time. Unfortunately, the
Java language does not make this noble goal very easy. Numerous hurdles of syntax and detail have to be overcome before the first experience with a living a main method, including concepts such as static methods, parameters, and arrays in
the signature; a statement to create the object ("new"); an assignment to a variable; the variable declaration, including variable type; a method call, using dot notation; possibly a parameter list. As a result, most textbooks typically either have to work their way through this forbidding list, and only reach objects somewhere around the fourth chapter;
or use a "Hello, world""-style program with a single static main method as the first example, thus not create an objects at all. With BlueJ, this is not a problem. A student can create an object and call its methods as the very first activity! Because users can create and interact with objects directly, concepts such as classes, objects, methods, and
parameters can easily be discussed in a concrete manner before looking at the first line of Java syntax. Instead of explaining more about this here, we suggest that the curi-ous reader dip into Chapter 1—things will quickly become clear then. An iterative approach Another important aspect of this book is that it follows an iterative style. In the
computing educa-tion community, a well-known educational design pattern exists that states that important con-cepts should be taught early and often. It is common, when in-troducing types, to give a full list of built-in data
types, or to discuss all available kinds of loop when introducing the concepts first, and at the same time provide complete coverage of all topics encountered. Our experience with text-books is that much of the detail is initially distracting, and has the
effect of drowning the im-portant points, thus making them harder to grasp. In this book we touch on all of the important topics several times, both within the same chapter and across different chapters. Concepts are usually introduced at a level of detail necessary for un-derstanding and applying to the task at hand. They are revisited later in a
different context, and understanding deepens as the reader continues through the chapters. This approach also helps to deal with the iterative approach. Looking at the first few chapters, teachers used to a more sequential introduction will be
surprised about the number of concepts touched on this early. It may seem like a steep learning curve. It is important to understand everything about these fundamental concepts will be revisited again and again throughout the book
allowing students to get a deeper understanding over time. Since their knowledge level changes as they work their way forward, revisiting im-portant topics later allows them to gain a deeper understanding overall. We have tried this approach with students many times. Sometimes students have fewer problems dealing with it than some long-time
teachers. And remember: a steep learning curve is not a prob-lem as long as you ensure that your students can climb it! Project-driven approach The introduction of material in the book is project and then providing an
exercise to apply this construct to solve a task, we first provide a goal and a problem at hand determines what kinds of solutions we need. As a conse-quence, language constructs are introduced as they are needed to solve the problem at hand determines what kinds of solutions we need. As a conse-quence, language constructs are introduced as they are needed to solve the problem at hand determines what kinds of solutions we need. As a conse-quence, language constructs are introduced as they are needed to solve the problem at hand determines what kinds of solutions we need.
are discussed in detail to illustrate the important concepts of each chapter. Using two very different example projects. This will hopefully serve to capture the reader's interest,
and also illustrate the variety of different contexts in which the concepts can be applied. We hope that our projects serve to give teachers good starting points and many ideas for a wide variety of interesting assignments. The implementation for all our projects is written very carefully, so that many peripheral issues may be studied by reading the
projects' source code. We are strong believers in learning by read-ing and imitating good examples. For this to work, however, it is important that the examples are well written and worth imitating. We have tried to create great examples are designed as open-ended problems. While one or more versions of each problem are discussed in
detail in the book, the projects are designed so that further extensions and improve-ments can be done as student projects. Complete source code for all projects are designed so that further extensions and improve-ments can be done as student projects. Complete source code for all projects are designed so that further extensions and improve-ments can be done as student projects.
many others is that it is structured along fun-damental software development tasks, not necessarily according to the particular Java language constructs. One indicator of this is the chapter titles such as "Primitive data types" or "Control structuring by fundamental development
tasks allows us to present a more general introduction that is not driven by intrica-cies of the particular programming language utilized. We also believe that it is easier for students to follow the motivation of the introduction, and that it makes much more interesting reading. As a result of this approach, it is less straightforward to use the book as a
reference book. Intro-ductory textbooks and reference books have different, partly competing, goals. To a certain extent a book can try to be both, but compromises have to be made at certain points. Our book is clearly designed as a textbook, and wherever a conflict occurred, the textbook style took precedence over its use as a reference book. We
have, however, provided support for use as a reference book by listing the Java constructs in-troduced in each chapter introduced in each chapter introduced
lines. Fixed bug with terminal sometimes not showing up in the documentation view on Windows. Fixed bug with rectangle characters showing up in the documentation view on Windows. Fixed bug with assert panel not changing the OK button to enabled (and the field to disabled) when selecting null/not-null as the assertion while
recording a test Show quick fixes even if the locale is one that supports localised Java compiler error messages (e.g. German) The Edit->New CSS/Text File and will let you create text files Select several lines in the Java editor, right-click and select "Screenshot line(s)". A screenshot of those lines will be put on the
clipboard. The lines must all be visible on screen. In the Java editor, middle-click the name of any other class to go to its definition/documentation. (Also available on right-click menu.) When you inspect a String object, new pane for string content and new "Copy to clipboard" button. New "Re-run" button on left side of main window, re-runs last
constructor or static method (or JavaFX app) with same parameters. (Compiles first, if needed.) The New Class dialog now has a new dropdown: Template or Minimal; the latter lets you create a class/interface/enum/record/etc without anything in the body. New checkbox in Preferences, in Interface -> General that lets you turn off re-opening last
open projects. Slightly increased the space for line numbers to help with four-digit line numbers on top" menu option. Made Cmd-comma open preferences on macOS. Ctrl-+ should increase font size on non-QWERTY keyboards. Increased default font size to 12 in new installations. Line numbers on by default in new
installations. Editor tabs can be dragged to reorder them in the editor windows. Ctrl-scrollwheel on Windows and Linux, and two finger pinch-zoom gesture on all OSes changes the editor font size. Changed the default Java editor and terminal font to "Source Code Pro", which supports box-drawing characters. 5.4.2 12 March 2025 Changes in this
version: Upgraded jGit library andswitched jGit-SSH library. Fixed various Git issues with SSH. Fixed Git "Check connection" to stop saying connection was fine in some cases when it was not. Changed Git settings dialog to ask for access token instead of password when connecting to Github/Gitlab. Removed macOS messages from Terminal window
MSI installer would allow multiple per-machine installations of different versions (in the same directory). BlueJ 5.4.1 will correctly remove all prior per-machine installations of earlier BlueJ versions (in the same directory). BlueJ 5.4.1 will correctly remove all prior per-machine installations of earlier blueJ versions. Fixed a missing message in the Stride editor. Fixed a bug where the right-click menu for the Java editor side bar (which let you turn on line numbers)
would not appear on Windows. 5.4.0 9 August 2024 Changes in this version: Moved to Java 21, including support for the new language feature switch patterns (including records). Changed Mac build to use the JDK for Apple silicon (M1, M2, etc); Intel build is still available. Added ARM64 Linux build to support Raspberry Pi. Fixed bug with scope
highlighting sometimes being missing on some lines. Fixed some issues with entering non-English characters (e.g. Chinese, Korean) in Java and Stride. Removed the method name tooltip on hover in the terminal not re-rendering when font size is
changed. Stopped infinite compilation which could occur when you have an error in a child class with a sealed parent. Fixed a bug where inheritance arrows for sealed classes could be double-headed. Fixed a bug where inheritance arrows for sealed classes could be double-headed. Fixed a bug where inheritance arrows for sealed classes could be double-headed. Fixed a bug where inheritance arrows for sealed classes could be double-headed. Fixed a bug where inheritance arrows for sealed classes could be double-headed. Fixed a bug where inheritance arrows for sealed classes could be double-headed. Fixed a bug where inheritance arrows for sealed classes could be double-headed. Fixed a bug where inheritance arrows for sealed classes could be double-headed.
2024 Changes in this version: Mac: Fixed splash screen not showing up until quite late, sped up startup process, fixed application Added JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load Stopped Terminal showing blank stderr pane when you run a JavaFX application not getting focus after load stderr pane when you run a JavaFX application not getting focus after load stderr pane when you run a JavaFX application not getting focus after load stderr pane when you run a JavaFX application not getting focus after load stderr pane when you run a JavaFX application not getting focus after load stderr pane when you run a JavaFX application not getting focus after load stderr pane when you run a JavaFX application not getting focus after load stderr pane when you run a JavaFX application not getting focus after load stderr pane when you run a JavaFX application not getting focus after load stderr pane when you run a JavaFX application not getting focus after load stderr pane when you
history due to sorting by 12-hour clock time Fixed Git history to only show files changed in that commit (rather than all files present in the repository then) Some slight speed-ups to Java editor performance Added code completion for local variable names (incl parameters, for-each loops and instanceof vars) Added section boundaries in the terminals,
to group together output from each method callm and added tooltip showing the method call that produced each section (even if you aren't logging method calls) Clear the stderr pane in Terminal on each method call Updated the Git library we use Changed backspace behaviour after if/else in Stride to only remove the else, not the whole if Fixed a
bug where Stride didn't save when you added/removed a default clause in a switch frame Fixed a bug which could prevent compiler errors being shown when a stale class file was present Fixed a bug where BlueJ would endlessly compile blank files Changed debugger split pane to be 3-way Stopped object highlight enlarging as you step through code
Added user-contributed Chinese translations 5.2.1 10 October 2023 A full list of bug fixes can be found on Github. Some highlights: Fixed the Linux installer not working outside Linux. Fixed the JavaFX warning not being suppressed in the terminal in non-English locales. Fixed
issue with permits clauses not being parsed correctly when after an extends or implements clause, and messing up dependency calculation. 5.2.0 20 June 2023 A full list of bug fixes can be found on Github. Some highlights: Fixed an issue on Mac where double-clicking in Finder to open a project would load/focus BlueJ but not open the project. Fixed
launch crash on Ubuntu. Fixed a bug where errors on the last line of a Java file (often, reached end of file while parsing after a missing close curly bracket) would not show at all. Removed annoying warning on stdout in the Terminal when running JavaFX programs. You can now focus classes in the class diagram by typing their name (e.g. typing "Tic"
will focus "TicketMachine"). Mac is now packaged differently, as a DMG. 5.1.0a 27 October 2022 This is a Mac-only release to fix the issue with crashing on load on macOS 13 (Ventura). 5.1.0 20 September 2022 In this release to fix the issue with crashing on load on macOS 13 (Ventura).
editor. 5.0.3 28 March 2022 In this release: Many different accessibility improvements, especially to the terminal, codepad, but also the general interface. Compiler errors can now be seen in a list pane when accessibility mode is turned on. Fixed a bug in saving the password for Git in the teamwork. Various small editor bug fixes, including one where
errors would sometimes not show in the editor. Added System.in/out/err to the code completion dialog as a special case. Stopped passing -source to the compiler by default (this now allows users to specify --release). Cut/copy now do nothing if the selection is empty (previously it would blank the clipboard). Added new method to the plugin API. 5.0.2a
2 December 2021 This is a Mac-only release to address a crash that was happened on MacOS 12 (Monterey). All other operating systems should continue to use 5.0.2. 5.0.2 6 August 2021 In this release: Fixed a problem where BlueJ would crash when launching on some Windows machines. Fixed a bug where on Windows JavaFX menus and dialogs
would not work correctly in user programs. Some minor bug fixes to the Java and Stride editors. Fixed a bug where sometimes red error underlines would not show up in the Java editor. Added new methods to the plugin API. A full list of fixes and improvements is available here. 5.0.1 30 April 2021 In this release: Many Java editor bug fixes, including
scrolling while drag-selecting, tab key behaviour, bracket highlighting, smooth scrolling on Mac. Added a new mechanism to support opening text files in the project in the BlueJ text editor. Switched to GDK 2 on Linux to
help with some windowing issues. Fixed some internal errors which could occur on compilation and cause errors not to be shown. A full list of fixes and improvements is available here. 5.0.0
28 January 2021 In this release: Rewrote the Java editor implementation to be faster and more robust. Added support for JUnit 5 and switched to JUnit 5 by default (JUnit 4 is still supported as before). Rewrote the extensions API to work with the new editor.
Removed the deprecated Subversion support. Bundled the JDK on Ubuntu to avoid some JavaFX dependency issues. A full list of fixes and improvements is available here. 4.2.2 4 October 2019 In this release: Fixed an issue with an extra space appearing in the terminal before the first output. Fixed not being able to alter the preferences for the
Submitter extension. A full list of fixes and improvements is available here. Note: Subversion support is now deprecated and scheduled for removal. Users are encouraged to migrate to using Git instead. Note: The extensions API is planned to change soon, so extension writers will need to make changes. Further details will be available in the coming of the coming of
months. 4.2.1 30 April 2019 In this release: Fixed a startup freeze that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows window positioning issue that could cause Blue windows. Fixed a window positioning issue that could cause Blue windows window positioning issue that could cause Blue windows window positioning issue that could cause Blue window positioning issue window positioning window positionin
when dismissed, and made it less likely to appear in cases which might be caused by slow network storage. Fixed an issue with not recognising JavaFX installed via Ubuntu package. A full list of fixes and improvements is available here. Note: Subversion support is now deprecated. Users are encouraged to migrate to using Git instead. 4.2.0 7
Feburary 2019 In this release: Moved to Java 11. This means that BlueJ now requires a 64-bit operating system. Added support for the new var keyword in Java 10. Fix JUnit BeforeClass/AfterClass functionality. Fix various small editor bugs. A full list of fixes and improvements is available here. Note: Subversion support is now deprecated. Users are
encouraged to migrate to using Git instead. 4.1.4 29 October 2018 In this release with support for Subversion. Users are encouraged to migrate to using Git instead. Note: See Version 4.0.0 notes for supported devices,
minimum requirements and major changes since Version 3.1.7 Note: This is the last version of BlueJ supported devices, minimum requirements and improvements is available here. Note: See Version 4.0.0 notes for supported devices, minimum requirements and improvements is available here. Note: See Version 4.0.0 notes for supported devices, minimum requirements and improvements is available here. Note: See Version 4.0.0 notes for supported devices, minimum requirements and improvements and improvements is available here. Note: See Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 4.0.0 notes for supported devices, minimum requirements and improvements is available here. Note: See Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 4.0.0 notes for supported devices and major changes since Version 4.0.0 notes for supported devices and major changes since Version 4.0.0 notes for supported devices and major changes since Version 4.0.0 notes for supported devices and major changes since Version 4.0.0 notes for supported devices and major changes since Version 4.0.0 notes for supported devices and major changes since Version 4.0.0 notes for supported devices and major changes since Version 4.0.0 notes for supported devices and major changes since Version 4.0.0 notes for supported devices and major changes since Version 4.0.0 notes for supported devices and major changes since Version 4.0.0 notes for supported device
requirements and major changes since Version 3.1.7 4.1.2 9 November 2017 Major points: Added: Interactive tutorial (under Help -> Interactive Tutorial). Fixed: Stale bugs displayed in editor / compile command ignored. Fixed: Git support issues, especially with sub-packages on Windows
A full list of fixes and improvements is available here. Note: See Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 3.1.7 4.1.1 18 September 2017 Major fixes: Fixed: Occasional exceptions/bugs during Java file editing. Fixed: Many printing problems. Fixed: Class documentation view shows blank on
first view. A full list of fixes and improvements is available here. Note: See Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 3.1.7 4.1.0 23 June 2017 Major fixes: Fixed: Graphical display bug could cause the Java editor and other windows (e.g. Terminal) to turn white and not redraw correctly. Fixed
Several teamwork issues and bugs (especially in Git). Fixed: Several controls (such as buttons in editor) could appear tiny on high-DPI screens on Windows. A full list of fixes and improvements is available here. Note: See Version 4.0.0 notes for supported devices, minimum requirements and major changes since Version 3.1.7 4.0.1 28 March 2017
Major fixes: Fixed: Closing code editor kills created objects and clears the the object bench Fixed: After switching to another language, Preferences disappear from the Mac app menu and some keyboard shortcuts will not work Fixed: Add
 «stereotype» markers again («/») in class diagram Fixed: Lack of scrolling with lots of methods either in unit test, Class popup or Object popup menus Fixed (MacOS X): Projects can't be opened with double click on "package.bluej" when BlueJ is open A full list of fixes and improvements is available here. Note: See Version 4.0.0 notes for supported
devices, minimum requirements and major changes since Version 3.1.7 4.0.0 8 March 2017 This is a major new release of BlueJ. For Java programming, it adds display of multiple errors and has most of the BlueJ interface rewritten into JavaFX (solving many HiDPI issues on Windows) while also
providing some support for writing JavaFX applications and including JavaFX applications and including JavaFX applications and stride. Support for writing JavaFX applications and including JavaFX applications and stride. Support for writing JavaFX applications and including JavaFX applications and stride. Support for writing JavaFX applications and including JavaFX applications and stride. Support for writing JavaFX applications and including JavaFX applications and including JavaFX applications and stride. Support for writing JavaFX applications and including JavaFX applications are support for writing JavaFX applications and including JavaFX applications are support for writing JavaF
support for the Git version control system (and retains support for Subversion). Note: Blue 4 requires Java 8 (minimum 8u60) and JavaFX, and thus is currently unsupported on older Macs and on Raspberry Pi. Support for CVS, JavaME projects, and applet development has been removed. Some extensions may no longer work in Blue 4. 3.1.7
February 2016 This is a bug-fix release for users who still need to use Java 6. In the absence of any major issues, it will be the last version of BlueJ which with Java 6. List of fixes and improvements: Fixed problems in handling SVN projects. Fixed minor issues with the debugger. Note that from this version, to work around bugs in the SVNKi
library that BlueJ uses for Subversion support, BlueJ projects which are shared to or checked out from a repository will use the (old) version command line client) may not be able usable with BlueJ projects. Projects checked out or shared
using BlueJ version 3.1.6 used the version 1.7 working copy format and will not work properly (in any version of BlueJ) due to SVNKit bugs; BlueJ version 3.1.6 22 October 2015 Minor bug fixes since last release. This will be the last version of BlueJ version 3.1.6 used the version 1.7 working copy format and will not work properly (in any version of BlueJ) due to SVNKit bugs; BlueJ version 3.1.6 used the version 1.7 working copy format and will not work properly (in any version of BlueJ) due to SVNKit bugs; BlueJ version 3.1.6 used the version 1.7 working copy format and will not work properly (in any version of BlueJ) due to SVNKit bugs; BlueJ version 3.1.6 used the version 1.7 working copy format and will not work properly (in any version of BlueJ) due to SVNKit bugs; BlueJ version 3.1.6 used the version 1.7 working copy format and will not work properly (in any version of BlueJ) due to SVNKit bugs; BlueJ version 3.1.6 used the version 3.1.6 used the version 3.1.7 will issue a warning when opening such a project. Note: This is the last version of BlueJ version 3.1.6 used the version 3.1.6 used the version 3.1.7 will issue a warning when opening such a project. Note: This is the last version and ve
of Blue which will work with Java 6. List of fixes and improvements: Fixed windows launcher in bundled build sometimes asking for JDK selection or reporting that no suitable JDK could be found Fix command-+ and command-- and c
improvements. Bug fixes and improvements: Fixed: Scope highlighting for various cases Improved detection of sudo permissions Added Raspberry Pi configuration tab Fixed: deleting text when a breakpoint is set causes an exception 3.1.4 26 September 2014 Support for Java 8 language features, plus various bug fixes and usability improvements
Bug fixes and improvements: Allow "get" to put package-private members on the object bench Uses system language by default (if a translation is available) Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited methods") Fixed: Missing class name in project documentation (under "inherited metho
projects available for checkout Fixed (version 3.1.4a on debian/raspbian only): Finding java JDK in latest Raspbian. 3.1.1 27 January 2014 Various bug fixes and usability improvements, especially for screen readers. Bug fixes: Fixed: class type parameters not shown in class diagram Fixed: issues with highlighting, pasting and deleting text in codepad
Added: message displayed when no project open, for clarification Fixed: Inspectors could not be moved by dragging them on Mac OS X Fixed: Debian package now recognises oracle-java7-jdk package 3.1.0 10 June 2013 Fixes for a few Mac bugs and the Portuguese language. Improved support for screen readers. Added data collection research
 project. Bug fixes: Fixed: Portuguese language selection not working Fixed: Mac OS X asks to install locations 3.0.9 22 February 2013 Interface language is now selectable from preferences dialog. Improvements to the display
exceptions in the codepad. New Mac OS X release which bundles JDK 7. Bug fixes: Fixed: Error when first creating class documentation on a project on a network share Fixed: Incremental parsing bug (scope highlighting while editing) Fixed: Debugger: thread controls
not enabled correctly for main thread Fixed: Mishandling of ArrayList in codepad/object bench Fixed: JUnit 4 support treats failures as errors Fixed: Disabling auto-indent glitches the closing bracket '}' Fixed: Auto-indent doesn't fire
after entering '{' at line beginning Fixed: Interactive invocation: Javadoc not shown for methods with parameters of a nested parameterized type 3.0.8 19 July 2012 Additions to the extensions (shown in the terminal window) are now coloured and clickable.
Resolves more firewall startup issues. More flexibility in bluej.userHome setting (possible to specify multiple locations). Bug fixes: Fixed: Parsing bug with multiple initializers in 'for' loop Fixed: Parsing bug with parenthesized identifiers Fixed: Parsing bug with multiple initializers in 'for' loop Fixed: "Maybe
you meant ... "misfires when there is more than one method call on a line Fixed: Update JDK search path (Ubuntu/Debian) 3.0.7 7 February 2012 Resolves firewall/network-adapter startup issues, especially on Max OS X 10.7 (Lion). Fixes a problem with inspecting arrays of two dimensions or more (and a similar problem with local multidimensional
arrays in the debugger). Fixes problem with non-ASCII characters in interactive method calls. Also fixes several other bugs. Bug fixes: Fixed: Class icons overlap when project is checked out without project files Fixed: Parser problems (bad highlighting and/or editor lockup) caused by annotations Fixed: Varargs parameters not treated as arrays by
code completion/error message enhancer Fixed: Search highlights wrong if file modified in external editor 3.0.6 11 November 2011 Resolves issue with Java 7. Introduces an .msi package for installation on Windows. Also fixes several bugs. Bug fixes: Fixed: Strange "CUI" icons in
editor window (Mac OS) Fixed: Missing space in error message Fixed: Windows launcher crashes if project path longer than 50 characters Fixed: Codepad rejects some syntactically valid expressions that have non-void types Fixed: Comment
attached to method drops first character when shown in call dialog Fixed: Interactively calling instance method with rull result fails to show result dialog Fixed: Codepad not casting some values correctly Fixed: Interactively calling varargs method with reference type parameter hangs 3.0.5 1 August 2011 Includes support for Java 7 language
features. JUnit 4 replaces JUnit 3 for test support (with thanks to Patrick Doran-Wu at the University of Western Australia). Default characters et for new projects changed to UTF-8 (better support for non-English characters). Improved method call recording in the terminal. Bug fixes: Fixed: Compile error could cause compiler to hange Fixed: One line
comments should not affect scope colour Fixed: Various small scope highlighting issues Fixed: Various minor problems in the code pad And many more minor bug fixes. 3.0.4 25 November 2010 This version is the last version of Blue which will work with Java 5. Bug fixes: Fixed: Inspectors not updated after codepad statement execution Fixed:
Resizing result inspector doesn't resize result field display Fixed: Cam't construct object with type parameters Fixed: Compile and code completion problems with code that extends Swing classes Fixed: Scope highlighting broken when inserting inner classes Fixed: Auto-layout sometimes inserts odd line
breaks 3.0.3 19 October 2010 Includes a Slovak translation, and enables assertions by default. Adds an editor function to add a javadoc comment to the current method. Bug fixes: Fixed: Inheritance arrow not drawn between classes that aren't in the default package Fixed: Some expressions generate "not a statement" error in the codepad Fixed:
"access$000" method when used immediately after compilation Fixed: Code completion fails on arrays of primitive types Fixed: Protected methods from grandparent do not shown in code completion Fixed: "Rebuild package" can crash with
ConcurrentModificationException if class name changed Fixed: It should not be allowed to draw inheritance arrows to/from Enums Fixed: Code completion throws UnsupportedOperationException in some cases Fixed: Array variables declared in codepad don't work if array declarators follow variable name Fixed: Expressions involving primitive-type
field access can cause compiler error in codepad Fixed: Printed source code line numbers are offset by 1 Fixed: Code completion on List expression fails Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes dialog sometimes displays "null" instead of class name Fixed: Windows: Using enter key to close method call dialog sometimes dialog sometimes
August 2010 Bug fixes: Fixed: Incorrect/unhelpful compiler error messages in the codepad Fixed: Using length on arrays fails in the codepad Fixed: Last character of bluej.vm.args setting is ignored Fixed: Compiler hangs when calling non-existing
method from some classes Fixed: Parser failure on Enum constants with bodies Fixed: Inspect doesn't work with array references 3.0.1 8 July 2010 Bug fixes: Fixed: Class templates for some languages (eg. Czech) were corrupted in Windows installation Fixed: Debian
package puts libraries in /usr/bin/bluejlib instead of /usr/share/bluej Fixed: Cannot re-create removed class Fixed: Cannot re-create removed class Fixed: Scope doesn't repaint if you de-indent a method which has a class variable before it Fixed: No comments
shown in call dialog when using certain parameters Fixed: HTML should have been generated in UTF-8 Fixed: Some expressions in the code pad are broken Fixed: HTML should have been generated in UTF-8 Fixed: Autolayout doesn't work Fixed: Autolayout d
respect tab size setting Fixed: Code completion within inner classes not working 3.0.0 30 May 2010 Summary of most important changes: Editor: Introduced code completion editor: Introduced scope highlighting Editor: Introduced functionality Look and
feel changes Editor: Improved matching bracket highlight Double-click on project.bluej file launches BlueJ and opens project Added context menu (right-click) in editor Javadoc shown for calls to methods with inner classes as parameters 'Print' option added in terminal 'Paste' function added in terminal Printing options saved across sessions Source
code printing can now include line numbers and syntax colouring OK/Cancel button ordering now follows OS custom on Mac OS Editor: word selection improvements. 3.0.0 preview 2 11 May 2010 Bug fixes and speed improvements. 3.0.0 preview 8 March 2010
First feature additions for version 3.0.0 2.5.3 8 October 2009 Summary of most important changes: Fixed: Java ME Deply button somtimes active for non-ME projects Fixed: Can't
close preferences dialog when toggling "show testing tools" in an empty package frame Fixed: Blue fails to start with Java 5 64bit on Mac OS Fixed: "Rebuild package" when unsaved class with changed name exists causes ConcurrentModificationException Fixed: On Windows, installation in a location with a non-ascii character in the path causes
launch failure Fixed: Subversion merge conflicts cause "update" dialog to spin forever (NullPointerException) Fixed: Teamwork status window stays open after project is closed 2.5.2 21 August 2009 Summary of most important changes:
Improved the installer so that Blue can be installed without adminstrator access rights on Windows. Double-click on a blue project file will launch it in Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type with Blue (on every OS but Mac OS you have to manually associate the file type w
window could hang the system. Fixed: Editing actions throw exceptions when in interface view. Fixed: Typing "5;" int to the codepad hangs 
version, Java version, Operating System, interface language etc) to the BlueJ maintainers, to help with development planning. This can be disabled by adding a "bluej.uid=private" setting to your bluej.properties file. Added some control over the date formatting by the Submitter extension Includes newer version of Svnkit library, should resolve some
subversion issues Fixed: "New Project", then selecting an existing project gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) sometimes gave a "MSVCR71.dll not found" error Fixed: Blue launcher (Windows) sometimes gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) sometimes gave a "MSVCR71.dll not found" error Fixed: Blue launcher (Windows) sometimes gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) sometimes gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) sometimes gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) sometimes gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) sometimes gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) sometimes gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) sometimes gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) sometimes gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) sometimes gave a bad error message Improved cursor behaviour in editor slightly Fixed: Blue launcher (Windows) slig
Windows OSes Fixed: Couldn't open jar file via "Open non-Blue]" 2.5.0 7 October 2008 Summary of most important changes: Added support for developing J2ME applications. Team work functionality now works with Subversion (SVN), as well as CVS. JUnit test recording significantly improved (records more interactions). Improved window handling.
Installer improvements. Numerous bug fixes. BlueJ now requires Java 5 or newer. Older Java versions are no longer supported. 2.2.1 17 December 2007 Summary of most important changes: Leaving breakpoints over compile, where possible. Added test class creation for applets. Adaptations for MacOS 10.5 (Leopard). User preference directory
location is now configurable. Fix bug with teamwork tools and packages. Make code pad font user-definable. Some other minor bug fixes. 2.2.0 6 July 2007 Summary of most important changes and packages. Make code pad font user-definable. Some other minor bug fixes. 2.2.0 6 July 2007 Summary of most important changes and packages.
Linux/Unix systems to Firefox Improvements of submitter extension: added https protocol and secure SSL/TLS mail transfer Print problems fixed Compiler not terminating in certain situations) fixed Compiler not terminating in certain situations fixed Compiler not terminating in unit tests and compiler not terminating in certain situations.
improved Freeze fixed related to self-referential object on bench Javadoc location can be changed with blue defendency chacking improved handling of objects of private inner classes 'userlib' and 'extensions' directory can be
specified in bluej.defs Printing \u000C' clears terminal 2.1.3 26 April 2006 Summary of most important changes: Minor bug fixes, including Shared memory transport fixed (Windows, as a failover for aggressive firewalls) Terminal encoding fixed (defaults to system encoding now) Terminal encoding can now be set with the bluej.terminal.encoding
property in bluej.defs Fixed: Mnemonics and accelerators don't work with escaped unicode sequences Fixed: README.TXT not saved on quit Fixed: BlueJ becomes unresponsive when program produces a lot of terminal
output Fixed: Leaving type argument blank for constructor calls doesn't work 2.1.2 10 February 2006 Summary of most important changes: Minor bug fixed bug in specification of VM location in bluej.defs. Bug fix in extension API: compiler warnings are now
propagated to listeners. 2.1.1 3 January 2006 Summary of most important changes: Minor bug fixes, most notably a bug that caused BlueJ to hang if source files ended in a comment without a terminating newline character. 2.1.0 25 November 2005 Summary of most important changes: The codepad now has support for declaring local variables which
retain their value between invocations. improvements for generics handling in codepad lots of bug fixes! 2.0.5 17 May 2005 Summary of most important changes: BlueJ now runs on Max OS X 10.4 (Tiger) improvements for generics handling in codepad lots of bug fixes! 2.0.5 17 May 2005 Summary of most important changes: BlueJ now runs on Max OS X 10.4 (Tiger) improvements for generics handling in codepad lots of bug fixes! 2.0.5 17 May 2005 Summary of most important changes: BlueJ now runs on Max OS X 10.4 (Tiger) improvements for generics handling in codepad lots of bug fixes! 2.0.5 17 May 2005 Summary of most important changes: BlueJ now runs on Max OS X 10.4 (Tiger) improvements for generics handling in codepad lots of bug fixes!
packages were shown in menu - fixed supplied fix for projects on UNC paths can now open jar files and zip files fix for browser opening on Windows ME 2.0.4 17 January 2005 Summary of most important changes: fixed bug that led to problems when installed in locations with space in path name 2.0.3 20 December 2004 Summary of most important
changes: #733 - hash in project path causes problems ("Main method does not appear in context menu"). #781 - keyword "new" highlighting #791 - inspection of generic field containing object of non-immediately-derived type #793/#795 - java 1.5.0 compile-time errors not verbose enough #800 - exception thrown in code pad causes internal BlueJ
error #838 - Japanese dialogs in wrong encoding #847 - auto project open at startup is not robust #848 - BlueJ main window now received focus at startup (MacOS) #786 - Terminal bug clearing error output 2.0.2 22 October 2004 Summary of most important changes: display for
unit test failures fixed systemLibrary property in bluej.defs fixed again use of alt/option key on MacOS fixed addition of memory based VM communication protocol (VM startup without tcp/ip) and corresponding bluej.defs property 2.0.1 4 October 2004 Summary of most important changes: invoking 'main' could hang virtual machine - fixed debugger
highlighting not updated correctly - fixed main method sometimes executes twice - fixed VM sometimes does not shot down correctly after reset - fixed 2.0.0 16 September 2004 Summary of most important changes: added support for J2SE 5.0 (generics, enums, etc) new diagram look new diagram handling functions: multi-selection, dragging/resizing
of multiple classes context menus for all diagram entities (including arrows) keyboard navigation of class diagram and object bench menu mnemonics improved extension API improved extension by default) Undo button affects inactive window.
applets don't get classpath jar bundle does not contain user lib jars Class inheritance arrows disappear when saving a project bench is not painted correctly Object bench is not painte
```

inspection BlueJ cant start with testing tools enabled Splitpane is not appropriately split remember preferred editor window position Remove command Unable to create virtual machine result inspector should be redesigned (show method call) key support for graph editing inspectors not properly updated Change the look of classes being moved. Change display of to null remote vm process does not quit when project closed too q... Mnemonics default applet template does not compile errors not shown properly Implementation arrows not selectable test result display should let you jump the line where an... Close terminal window with its project clear input buffer at execution start Multiple dialogs when deleting package BlueJ opens file twice click in selectino in editor does not unselect Javadoc for the Java API doesn't work Unicode character not supported in java source code BlueJ fails to launch browser ESC should close dialogs extra javascript in javadoc interface view Main thread

never terminates if another thread tries to j... The auto-indent should work in both directions There should be an opposite to TAB key indention 1.3.5 11 December 2003 Summary of most improvements (using VM optimisation by default) JUnit testing: auto-generated code improved auto-open of last used projects on startup System.exit now restarts VM improvements to extension API standard preference handling on MacOS number of 'recent projects' user definable startup without additional DOS window on MS Windows class diagram supports multiple selection changed inspector look better error messages easier addition of user libraries through use of 'userlib' directory colours for different class types user definable added missing label in some foreign languages 'VM optimize' now available in preferences new menu shortcuts for 'page setup', 'generate documentation' text input wirh multiple projects now works (one terminal per project) editor maintains last display state (interface/implementation) between sessions each package now has separate README note JApplet class now used by default when creating applets instead of Applet 1.3.0 7 August 2003 Summary of most important changes; more robust debugger/execution architecture extension API added submission extension added unit test support added improved diagram printing user definable scaling of diagram printout now using jdk 1.4.1 on MacOS if installed Agua interface improvements on MacOS web browser start problem on Win98/ME resloved EXIT ON CLOSE option bug fixed some documentation generation bugs fixed new interface languages available: Portuguese source print font size user definable printing form feed character (unicode 0x000C) now clears terminal terminal gets automatically cleared on recompile inspection of static fields now in class menu compiler warning display improved 1.2.2 16 October 2002 Summary of most important changes: freeze bug with jdk1.4.1 and debugger fixed 'open recent' option added to Project menu support for 'standout' comments System.err output in terminal improved project documentation with spaces in path works now evaluation of expressions added 1.2.1 August 2002 Summary of most important changes: bug with frequent access of floppy drive fixed bracket matching in editor implemented bluej.exe accepts parameters System.err output now directed to BlueJ terminal 'auto-clear' option in bluej terminal new interface languages: Chinese, Czech, Afrikaans 1.2.0 4 April 2002 Summary of most important changes: 326, 335 - support of jdk 1.4 added 285 - return key now activated OK button in parameter dialogs 330 - project documentation fixed with jdk 1.4 343 improved auto-indentation in editor 347 - last used project directory saved across sessions 000 - added windows installer 1.1.6 16 January 2002 Summary of most important changes: 166 - debugger now displays static variables in a separate list 166 - debugger displays variables when stopped in static method 225 - projects can include project-specific libraries 261 - Frames/JFrames automatically removed when objects are removed 235 - help menu items can be added by users 267 - terminal font size user definable 276 - javadoc can be configured to include private methods 295 - automatic creation of crash recovery files; optional creation of backup files (thanks to David R. Musicant (dmusican@mathcs.carleton.edu) for implementing these features) 241 - proper find/replace implemented in editor 303 - method template user definable 310 - japanese localisation added (interface, etc. in japanese); thanks to Runrun 275 - use Command key instead of CTRL as menu shortcut on MacOS 277 - bug fix on MacOS: Show Terminal/Debugger could crash system 298 - extended latin characters (umlaut, accented characters, etc) handled correctly by parser 1.1.5 24 September 2001 Summary of most important changes: 240 - documentation needs update (1.3 reg) 12 font size in editor not user definable 234 - Screen maximize 247 - cannot create jar file in project directory 248 - Some applet classes are not recognised when project is loaded 252 - Exported jar files containing packages unusable under Windows 255 - jikes error messages aren't shown 191 - Unexpected characters in input cause exception 1.1.4 14 June 2001 requires JDK 1.3 or newer uses new (JDK1.3) compiler implementation can get arrays and array element to the object bench local variables of static methods now shown (bug fix) new class templates can be added class templates can be located in arbitrary directory "user home" can be specified display of exceptions in library classes improved bug fixes with editor key definitions new editor functions for cutting/copying/moving lines and words optional soft TABs (tabbing with spaces) in the editor various bug fixes 1.1.3 29 March 2001 can now create objects of library types (eg. String) and call static methods breakpoints in main did not work - fixed editor tab size definable in bluej.defs file syntax colours user definable bug with interface views and multiple drives on Win32 fixed can now re-display project documentation without re-generating debugger "terminate" fixed object inspection from debugger fixed export for packages with sub-packages fixed 1.1.2 9 February 2001 added support for installation on MacOS X interface view of classes changed to display javadoc format in BlueJ editor inspection of elements of large arrays fixed System.exit now terminates all threads bugs with parsing explicit positive ints fixed more compiler error message help texts added "describe" function (CTRL-D) in editor fixed (had bug on Windows) static data now gets reinitialised with each call to "main(String[])" 1.1.1 1 September 2000 applets are now properly recognised even if not directly inheriting from "Applet" editor key bindings are now properly saved between sessions (IDK 1.3 only) "spinning barber pole on Linux fixed classes in the project are now properly recognised when source is not there (only .class file). This may be used to supply use-only classes to students without source bug with incorrect project closing fixed class positioning when adding class from file corrected long error messages in editor now wrap over two lines. Pressing the "?" button shows the complete message (together with an explanation) default argument for "public static void main(String[])" changed from null to { } to ensure compatibility with DOS editors have icons when minimised 1.1 4 August 2000 UML-style notation for the class diagram support for named and nested packages improved import of non-BlueJ packages terminal text can be saved or copied editor key bindings can be changed (jdk 1.3 only) editor font can be changed "export" function to create standard Java packages and executable jar files projects cascading menus for large menus the debugger can now be used with swing applications classes can be renamed in the class source 1.0.3 23 May 2000 static methods on abstract can be called now bug with calling methods after using "Get" has been fixed bug with printing the class diagram has been fixed bug with printing the class sources has been fixed/improved editor views have been improved bug fixed: end of long lines was not displyed properly bug fix: System.exit() did not work correctly new editor functions: "indent", "break-and-indent", "comment" editor key bindings can be changed (but are not yet saved - they are lost on exit) a default source directory can be defined in the "bluej defs" file. This serves as a starting point for the file chooser install support for jdk 1.3 added 1.0.1 17 January 2000 installation support for Linux jdk 1.2.2 added added preferences dialogue editor font size can be changed non-BlueJ packages can now be opened in BlueJ (with automatic dependency analysis). So far, only packages without nested packages are supported "packages..." line automatically fixed when importing classes System.exit() now works for graphical applications bug fixed in inspection of large arrays default parameter "null" added for static void main calls support for internationalisation (so far, German is the only language file supplied besides English) 1.0 23 August 1999 dependencies (eg inheritance relationships) are automatically entered into the source code if entered graphically several bugs in the debugger have been fixed 0.9.7 11 August 1999 text terminal reimplemented; fixed many problems terminal comes to front on output inspect for local variables in debugger implemented dependency display improved: changes in code immediately reflected in diagram 0.9.5 7 July 1999 Implementation changed to run on Java 2 platform Debugger implementation changed to use JDI - promises significant improvements in stability and reliability Applets supported (development and viewing in applet viewer and browser) Dependency analysis fixed - works correctly now Many editor improvements Help button for compiler error messages and exceptions Array inspection implemented Exception handling (reporting) improved 0.9.2a 12 May 1999 One bug fix: bluej could not be started on Unix systems without a command line argument. 0.9.2 5 May 1999 internal implemented - lists all user functions and their bindings many bug fixes 0.9.1 28 April 1999 printing of source files implemented cleaned up object popup menus classes can no longer be moved out of the screen removed bug in removing arrows backup package file now named "bluej.pkh" instead of "bluej returned properly bug fix: user defined object as method result now shown correctly bug fixes: names of objects on bench: now ensures that names are unique and valid bug fix: layout of object bench through the "Get" operation in the inspect window. This works now. 0.8.9 2 March 1999 Removing of classes has been implemented In the installer, a path to the java executable can be specified and will be used In multi-user system, where the BlueJ home directory is not writable to users, the system should now work much better. The Save As function has been implemented. Many other bug fixes... A free Java Development Environment designed for beginners, used by millions worldwide. Find out more... "One of my favourite IDEs out there is BlueJ"—James Gosling, creator of Java. Created by Supported by Download and Install Version 5.5.0, released 3 June 2025 (Many feature improvements, see more) Note: BlueJ requires a 64-bit operating system, which 95+% of users will have. For 32-bit operating systems, download BlueJ 4.1.4 instead. "Objects First with Java: A Practical Introduction Using BlueJ" is a textbook co-written by the developers of BlueJ and has sold hundreds of thousands of copies worldwide. Our free Blueroom website offers teaching resources and the chance to engage with other educators using Blue around the world. Tutorials and reference materials for working in Blue were be averall extensions available that add to Blue share functionality.

• fecamu

• http://activesites.atiks.org/files/95564d0b-bc19-4792-bc32-06877e184d4c.pdf

 $\bullet\ http://dhins.com/testingsites/advantage_aviation/assets/media/file/1f4a2022-18f9-4ab8-bcea-7b5d2551ff8e.pdf$

what is static electricity in simple wordsunit 9 progress check mcq ap chem

• horrible bosses 2 ending explained

hebuga
https://krygina40.ru/UserFiles/File/62081109377.pdf