

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра "ПОВТ и АС"

Знакомство с Java SE SDK

Методические указания к лабораторной работе
по дисциплинам "Программирование", "Объектно-ориентированное
программирование"

Ростов-на-Дону 20 г.

Составитель: к.ф.-м.н., доц. Габрельян Б.В.

УДК 512.3

Знакомство с Java SE SDK: методические указания – Ростов н/Д: Издательский центр ДГТУ, 20 . – 8 с.

Рассмотрены вопросы инсталляции средств разработки Java-программ, приведены примеры компиляции и запуска на выполнение приложений Java, создания Java-архивов и собственной документации. Даны задания по выполнению лабораторной работы. Методические указания предназначены для студентов направлений 090304 "Программная инженерия", 020303 "Математическое обеспечение и администрирование информационных систем".

Ответственный редактор:

Издательский центр ДГТУ, 20

1. Загрузка и инсталляция Java SE SDK и документации.

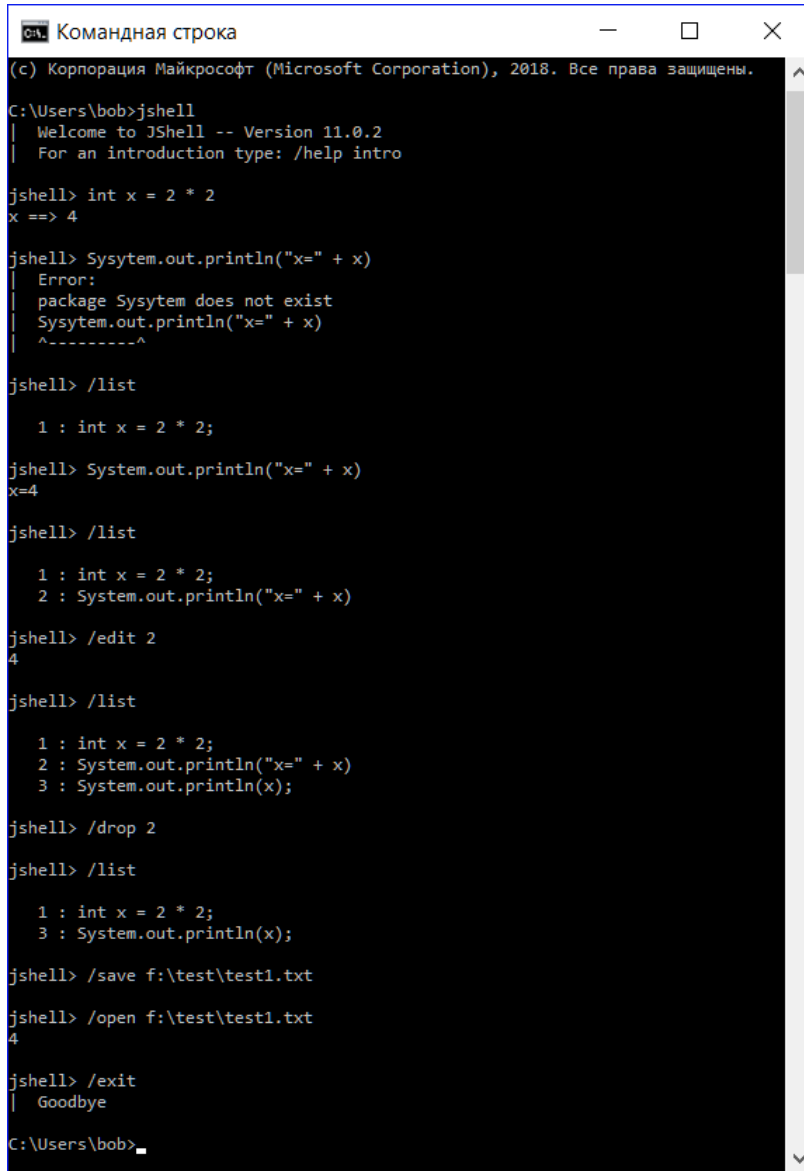
Java SE SDK это набор инструментов для разработки и запуска на выполнение Java-приложений. SDK и стандартную документацию можно скачать с сайта фирмы Oracle: <https://www.oracle.com/java/technologies/javase-downloads.html>. Использовать SDK в некоммерческих целях можно бесплатно, для коммерческого использования требуется платная подписка. Бесплатные версии распространяются несколькими разработчиками. Например, фирма Oracle предлагает Open JDK: <https://jdk.java.net>, сообщество группы пользователей Java и разработчиков, и поставщиков Java-продуктов - Adopt OpenJDK: <https://adoptopenjdk.net/index.html>. На сайте Oracle содержатся также руководства по языку и библиотекам Java (Java SE 8): <https://docs.oracle.com/javase/tutorial/>. Библиотеку элементов графического пользовательского интерфейса JavaFX и соответствующую документацию можно скачать на сайте Gluon: <https://openjfx.io>.

ЗАДАНИЕ 1. Найдите на локальном диске каталоги, содержащие Java SDK и стандартную документацию. Из подкаталога с документацией docs загрузите в браузер файл index.html. Посмотрите, как организована документация. В дальнейшем ее можно использовать в качестве справочного материала по всем пакетам, классам, интерфейсам и функциям стандартных библиотек Java SE. Наберите в строке поиска SEARCH (в правой верхней части окна) строку String. Посмотрите описания конструкторов и методов этого класса. Чем метод isEmpty отличается от метода isBlank?

2. Компиляция и запуск на выполнение Java-приложения.

Все инструменты SDK содержатся в каталоге /bin/. Если в переменной окружения процесса PATH задан путь к этому подкаталогу, то запускать SDK-программы можно из любого каталога. Кроме того, можно создать переменную окружения CLASSPATH с

указанием каталогов, содержащих библиотеки (пакеты) Java API. Задание CLASSPATH не является необходимым для простых примеров в этой лабораторной работе, но может понадобиться в дальнейшем.



```
Командная строка
(c) Корпорация Майкрософт (Microsoft Corporation), 2018. Все права защищены.

C:\Users\bob>jshell
| Welcome to JShell -- Version 11.0.2
| For an introduction type: /help intro

jshell> int x = 2 * 2
x ==> 4

jshell> Sysytem.out.println("x=" + x)
| Error:
| package Sysytem does not exist
| Sysytem.out.println("x=" + x)
| ^-----^

jshell> /list

  1 : int x = 2 * 2;

jshell> System.out.println("x=" + x)
x=4

jshell> /list

  1 : int x = 2 * 2;
  2 : System.out.println("x=" + x)

jshell> /edit 2
4

jshell> /list

  1 : int x = 2 * 2;
  2 : System.out.println("x=" + x)
  3 : System.out.println(x);

jshell> /drop 2

jshell> /list

  1 : int x = 2 * 2;
  3 : System.out.println(x);

jshell> /save f:\test\test1.txt

jshell> /open f:\test\test1.txt
4

jshell> /exit
| Goodbye

C:\Users\bob>
```

Если у Вас установлен SDK версии 9 или выше, для работы с небольшими наборами команд можно использовать интерпретатор команд Java (не JVM) jshell. jshell запускается в командной строке и позволяет вводить и выполнять Java-код построчно или загружать (и сразу же выполнять) его из указанного файла. У оболочки есть свои команды, которые начинаются со значка /. Посмотреть все эти команды можно набрав /help. С помощью /list можно просмотреть весь набор введенного кода. /drop удаляет указанный код. /save сохраняет введенный код в указанном файле, а /open загружает из файла и выполняет. Можно

отредактировать указанную введенную прежде строку кода с помощью команды /edit. Выход из оболочки - команда /exit. Руководство пользователя по jshell от Oracle (для Java 15) можно скачать по ссылке <https://docs.oracle.com/en/java/javase/15/jshell/java-shell-user-guide.pdf>.

Оболочка jshell не предназначена для серьезной разработки. Для этого нужен компилятор Java-программы в коды виртуальной машины Java (JVM). Компилятор - это программа javac.exe. Компилятору необходимо указать файл (или файлы) с программой и расширение java. Например,

javac Hello.java или

javac *.java компилировать все файлы с расширением java из текущего каталога
Если трансляция программы прошла успешно, должен появиться файл Hello.class (если программа Hello.java содержит только один класс, а противном случае для каждого класса будет создан свой файл с расширением class).

Для выполнения откомпилированной в коды JVM программы необходимо запустить интерпретатор JVM. В SDK это программа java.exe. Имя файла программы (class-файл) передается интерпретатору через параметр командной строки (без расширения!). Например,

java Hello

Управлять работой компилятора и интерпретатора можно с помощью опций (ключей), задаваемых в командной строке. Посмотреть все опции можно с помощью ключа -help.

javac -help

java -help

Версии компилятора и интерпретатора можно получить, запустив их с ключом -version.

ЗАДАНИЕ 2.

- 1) Посмотрите, какие значения заданы переменным окружения PATH и CLASSPATH.
- 2) Определите версии компилятора Java и интерпретатора команд JVM, установленных в Вашей системе.

3) Используйте любой текстовый редактор для создания файла Hello.java со следующим содержимым:

```
public class Hello {  
    public static void main(String[] s) {  
        System.out.println("Hello World!");  
    }  
}
```

Обратите внимание на то, что имя файла с исходным кодом программы должно в точности (с учетом регистра символов) совпадать с именем класса, а расширение должно быть java.

4) Откомпилируйте файл и запустите на выполнение программу. Вы должны увидеть в консольном окне (или окне терминала)

Hello World!

В Windows создайте файлы compile.bat для компиляции и run.bat для запуска произвольной Java-программы. Поместите в эти файлы команду set CLASSPATH=.;%CLASSPATH% если переменная CLASSPATH определена, но текущий каталог (.) в ней не указан.

5) Если у Вас версия Java 11 или больше, такую простую программу можно выполнить без явной трансляции. Удалите откомпилированный файл Hello.class и выполните

```
java Hello.java
```

3. Использование интегрированных сред разработки (IDE).

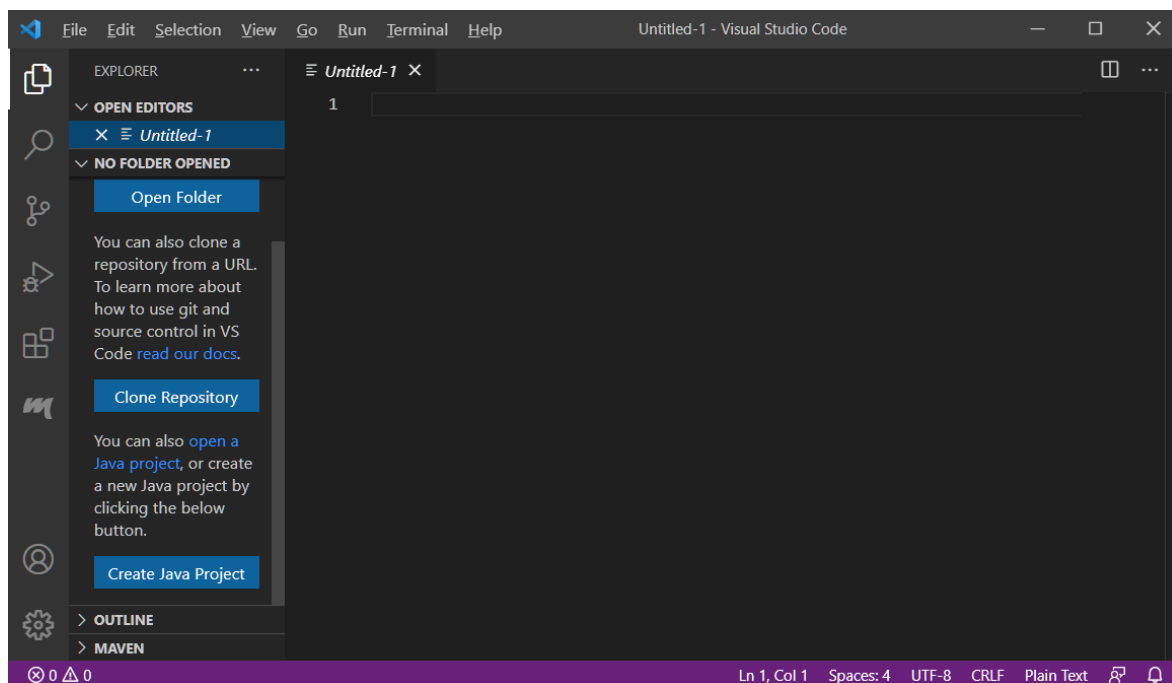
Существует немало тяжеловесных IDE, ориентированных на разработку сложных проектов на разных языках программирования. Для Java разработки самыми популярными являются IDE IntelliJ Idea компании JetBrains, NetBeans от Oracle и Eclipse

от организации eclipse. NetBeans и Eclipse бесплатные, IntelliJ Idea имеет бесплатную версию Community Edition. Создание проектов в этих средах интуитивно понятно.

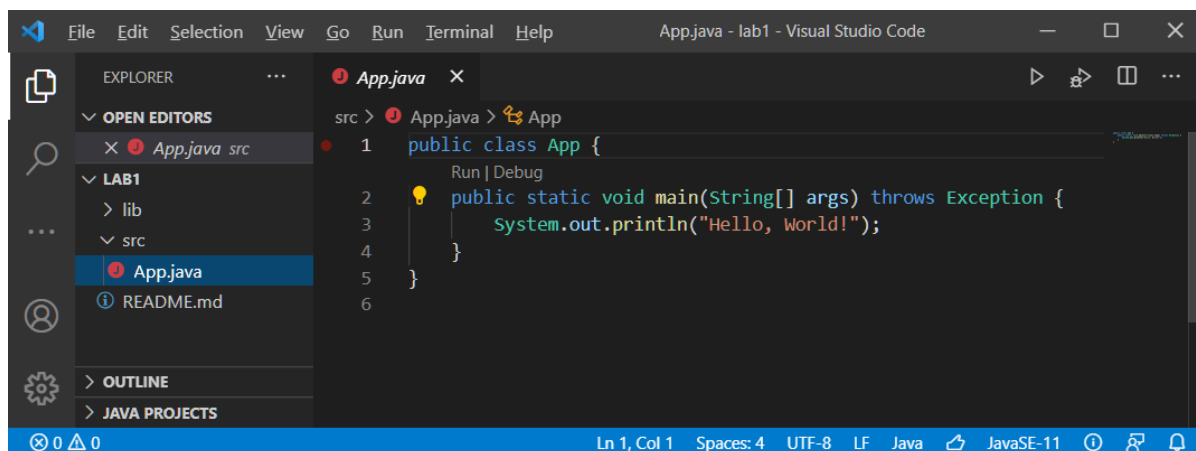
Можно использовать также более простые редакторы кода, например, Microsoft Visual Studio Code.

3.1. MS Visual Studio Code.

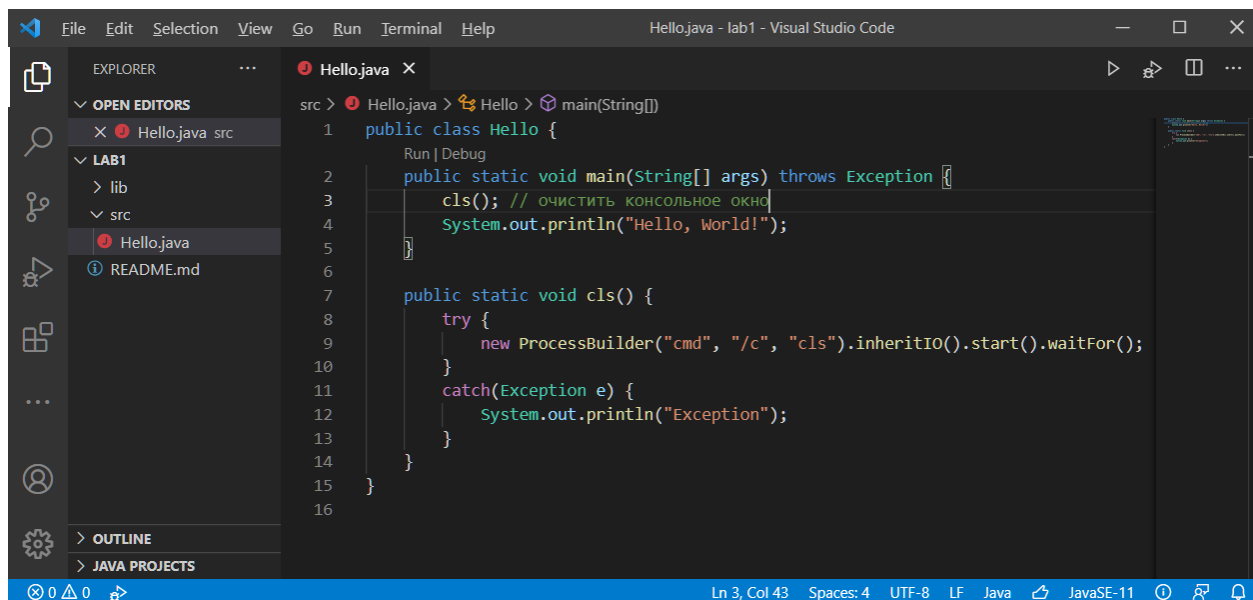
Есть версии для Windows, Linux, MacOS. Чтобы была поддержка проектов на каком-то языке программирования, нужно установить расширения редактора. Далее эти расширения можно настроить вручную. Но, для Windows (и MacOS) есть возможность скачать и установить (бесплатно) как единое целое набор программ Coding Pack for Java, содержащий собственно редактор VS Code, Java SDK и расширения для работы с Java в редакторе (<https://code.visualstudio.com/docs/java/java-tutorial>). После установки и запуска программы (файл code.exe в Windows) комбинация клавиш Ctrl+Shift+E открывает вкладку Explorer для управления проектами (для простых программ можно не создавать проект).



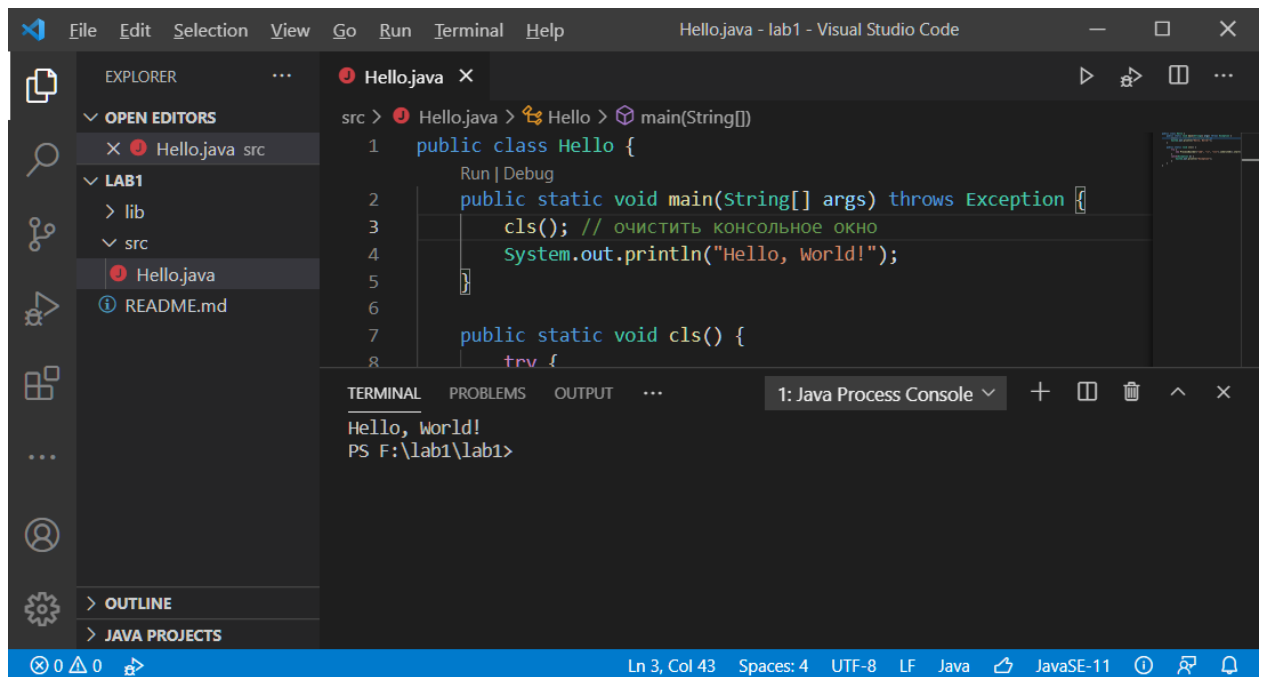
Далее выбираем Create Java Project, No build tools. Указываем папку, в которой будет размещаться проект. Задаем имя для проекта (например, lab1). Во вкладке Explorer открываем папку src. По умолчанию здесь будет располагаться автоматически созданный файл App.java.



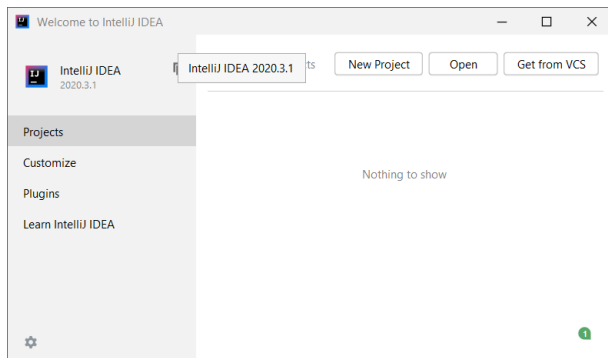
Щелкаем правой кнопкой мыши по App.java и переименовываем в Hello.java. Переименовываем также класс App в Hello. Добавим статический метод cls для очистки окна терминала (работает только в Windows).



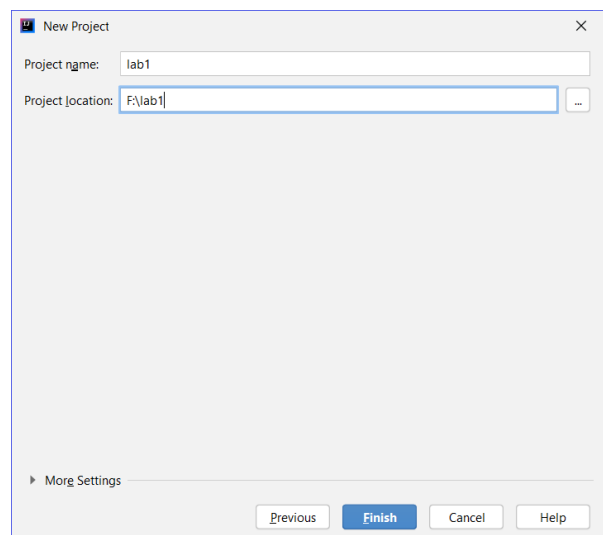
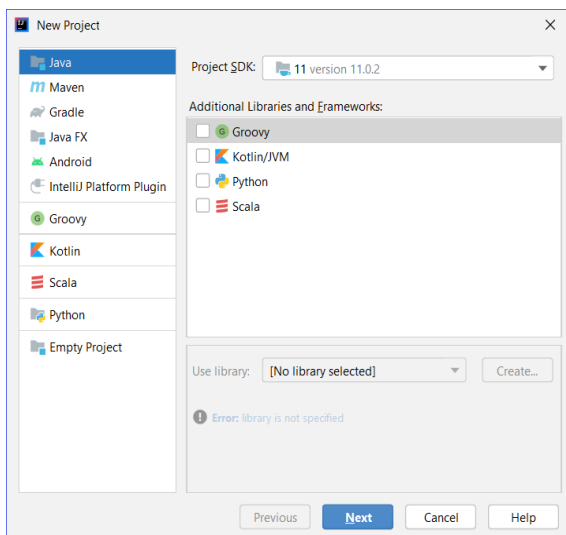
Компилируем и запускаем (Run над методом main или треугольник в верхнем правом углу или Ctrl+F5 ...).



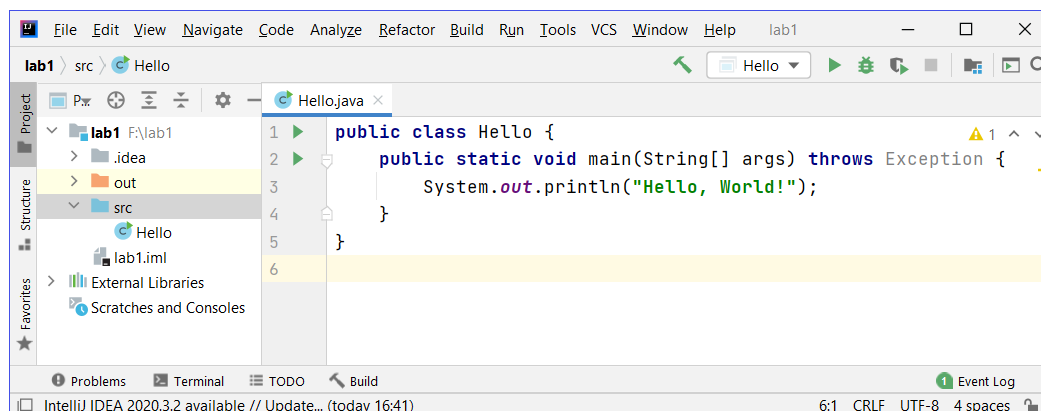
3.2. JetBrains IntelliJ Idea.



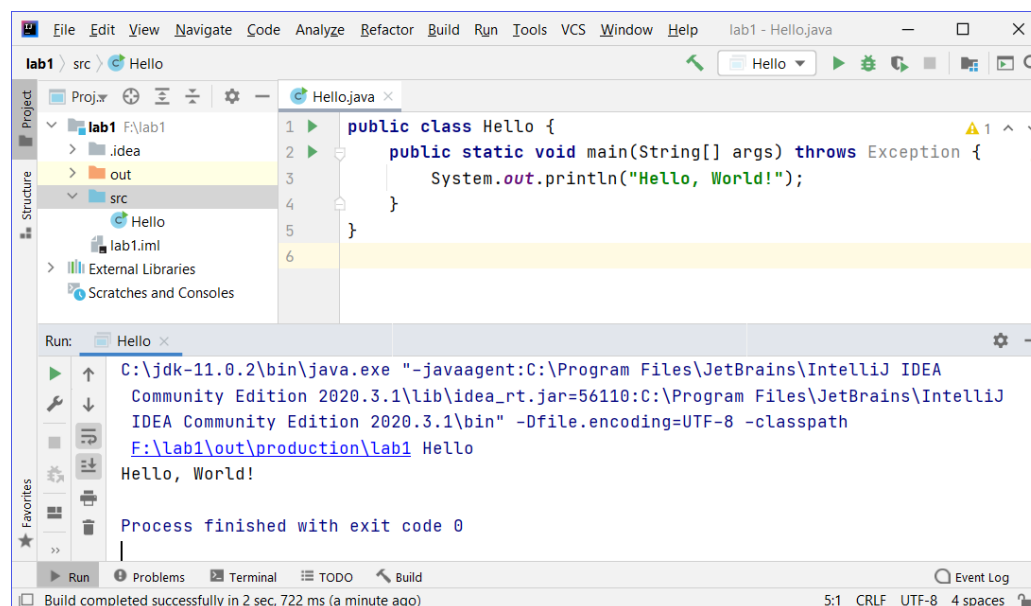
Скачать с сайта <https://www.jetbrains.com/idea/>.
В Idea, после запуска программы (idea64.exe),
выбираем New Project - Java - SDK по
умолчанию (можно выбрать один из
нескольких, если у Вас установлены разные



версии Java SDK), затем Next, отметить галочку возле Create project from template, Next, задаем имя проекта и папку, в которой он будет располагаться, очищаем поле Base package. После Finish создается класс Main, щелкаем правой кнопкой мыши по имени класса и выбираем Refactor-Rename, задаем имя Hello. Набираем текст программы (метод `cls` не работает).



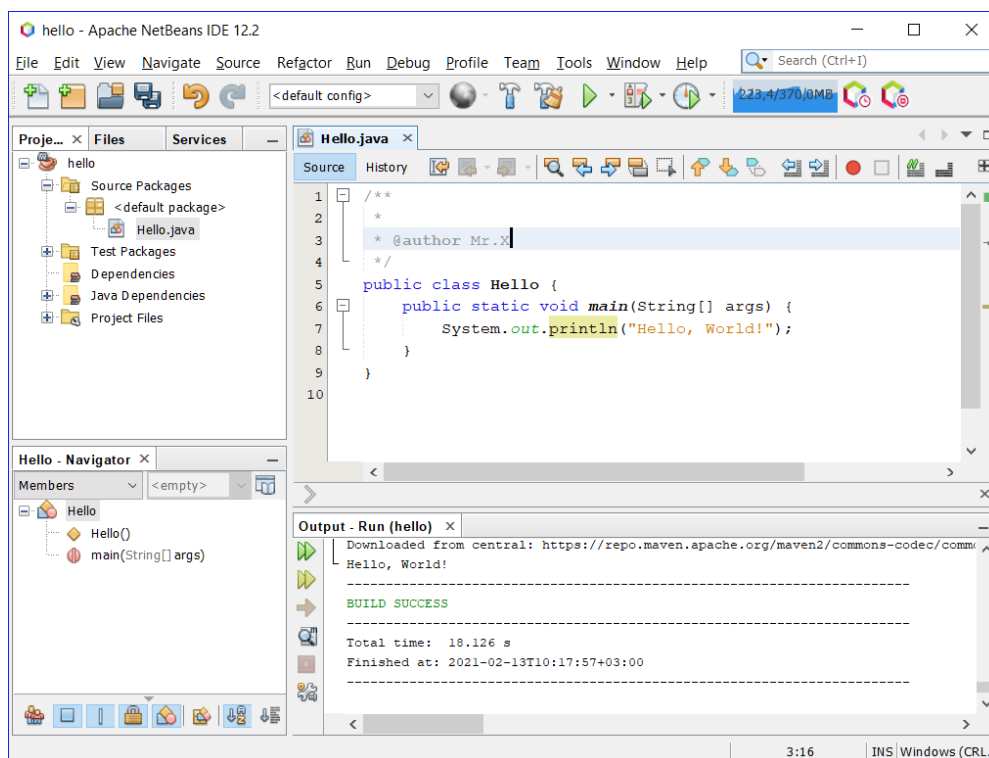
Запуск на выполнение `Alt+Shift+F10`. Выбираем Hello.



3.3. Netbeans.

NetBeans можно скачать по ссылке <https://netbeans.apache.org/download/index.html>. Создаем проект. New Project - Java with Maven - Java Application, Next. Может быть

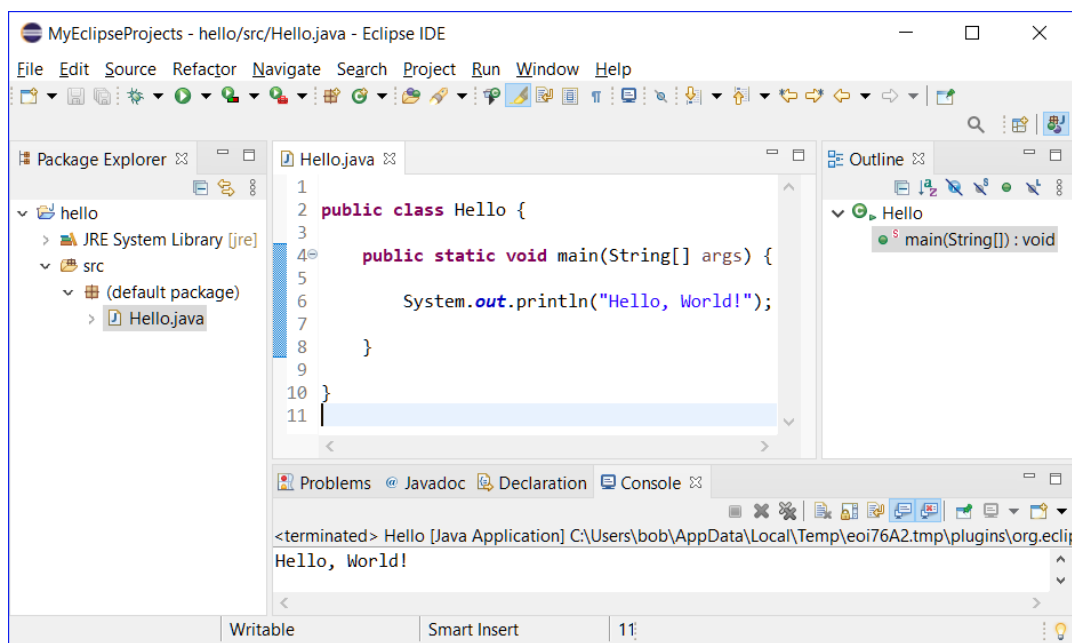
запрос на установку дополнительных модулей (nb-javac API). Отмечаем и нажимаем Download and activate ... Задаем имя проекта, его расположение, Group id, и версию. Пакет в данном примере можно не задавать (пустое поле). Finish. Projects-<default package> правой кнопкой New - Java class. Задаем имя класса Hello, остальное по умолчанию. Finish. Получаем заготовку для класса Hello с комментариями. Меняем комментарии и дописываем метод main в классе Hello. Создать проект (Build Project) - пиктограмма в виде молотка, через меню Run или F11. При первом запуске нужно иметь соединение с Интернет, будет автоматически установлен модуль для системы сопровождения проектов Maven. Запускаем на выполнение (через меню или зеленый треугольник на панели инструментов в верхней части окна или F6). Select Main Class.



3.4. Eclipse.

Eclipse можно скачать по ссылке <https://www.eclipse.org/downloads/>. В инсталляторе выбираем Eclipse IDE for Java Developers.

Создаем проект. File - New - Java Project. Задаем имя проекта hello, его расположение, на вкладке JRE выбираем Use default 'jre' and workspace compiler preferences. Finish. Create `mo.java` - оставляем по умолчанию. Don't Create (!!!). Правой кнопкой New - Class. Оставляем Source folder по умолчанию, делаем Package пустым, задаем имя класса Hello, which method stubs would you like to create - ставим галочку рядом с `public static void main(String[] args)`, остальное по умолчанию. Finish. Получаем заготовку для класса Hello с комментарием `// TODO Auto-generated method stub`. Удаляем комментарий и дописываем на его месте `System.out.println("Hello, World!");`; Запускаем на выполнение (через меню или зеленый треугольник на панели инструментов в верхней части окна или `Ctrl+F11`). В окошке Save and launch нажимаем Ok.



4. Создание программы с графическим пользовательским интерфейсом.

Изначально язык Java предлагал библиотеку компонентов для графического пользовательского интерфейса - Abstract Window Toolkit (AWT). Она давно устарела, но новые библиотеки используют ее возможности, в первую очередь для связи с элементами интерфейса конкретной операционной системы. Главное окно программы в этой библиотеке представлено классом `Frame`. В окне могут размещаться другие

элементы, например, текстовые метки. Кроме того, с окном могут быть связаны события. Например, событие от мыши если при нахождении курсора мыши в области окна щелкнуть ее кнопкой. Чтобы программа (окно) могла отреагировать на событие в программном компоненте, с которым связано событие (окне), нужно зарегистрировать некоторый код, который и будет вызван при возникновении этого события. Этот код называется обработчиком события. Обработчик события - это некоторый класс, в котором заданы специальные методы. В терминологии Java обработчики событий называются слушателями (Listener). С окном могут быть связаны специальные, «оконные» события - раскрыть окно на весь экран, минимизировать окно, закрыть окно. В примере ниже создается класс-наследник класса AWT Frame. Тем самым он сам является окном. В нем размещаются две метки (объекты класса Label) - firstLabel и secondLabel. Первая метка содержит неизменяемый текст “Hello my”, а вторая текст, который будет меняться, если щелкать мышью по поверхности окна (но не по областям окна занятым метками!). Эти значения выбираются случайным образом из массива строк text. Это делается в обработчике (слушателе) событий от мыши, зарегистрированном в окне. Кроме того, в этом же окне регистрируется слушатель оконных событий для обеспечения возможности закрытия окна (и тем самым программы в целом).

ЗАДАНИЕ 3. Создайте файл HelloFrame.java в текстовом редакторе или в доступной IDE, откомпилируйте и запустите на выполнение программу. Разберитесь в общих чертах с тем, как она работает.

```
import java.awt.*;
import java.awt.event.*;
import java.util.Random;

/** Simple Frame with Label
 * @version 1.0
 * @author Mister X
 */
public class HelloFrame extends Frame {
```

```

/** Program entry point */
public static void main(String[] args) throws Exception {
    HelloFrame hello = new HelloFrame();
    hello.setTitle("Hello!");
    hello.setBackground(Color.black);
    hello.setLayout(new FlowLayout());
    hello.setLabelAttributes(firstLabel);
    hello.setLabelAttributes(secondLabel);
    hello.add(firstLabel);
    hello.add(secondLabel);
    hello.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            Random rnd = new Random();
            int idx = rnd.nextInt(4);
            secondLabel.setText(text[idx]);
        }
    });
    hello.addWindowListener(new WindowAdapter() {
        @Override
        public void windowClosing(WindowEvent we) {
            System.exit(0);
        }
    });
    hello.setBounds(100,100,260, 80);
    hello.setVisible(true);
}

/** method that changes Label attributes
@param label - concrete Label
*/
private void setLabelAttributes(Label label) {
    Font font = new Font(Font.MONOSPACED, Font.PLAIN, 22);
    label.setFont(font);
    label.setBackground(Color.BLACK);
    label.setForeground(Color.YELLOW);
}

/** field - reference array of strings */
private static String[] text = {"Word", "Lord", "Nord", "Sword"};

```

```

/** field - Label for preamble */
private static final Label firstLabel = new Label("Hello my");

/** field - Label for variants of answers */
private static final Label secondLabel = new Label(text[0]);
}

```

4. Упаковка программ в Java-архивы.

Если проект состоит из нескольких файлов (с исходными кодами, с ресурсами), то удобно собрать их все вместе так, чтобы весь проект можно было представить одним файлом. Это можно сделать, поместив все нужные файлы в специальный архив.

Java SDK содержит программу-архиватор `jar.exe`, с помощью которой можно создавать файлы с расширением `jar`. Эта программа сжимает файлы по тому же алгоритму, что `zip`-архиватор. (Т.е. можно переименовать `jar`-файл в `zip`-файл и распаковать его с помощью `zip`-разархиватора). Запустив программу `jar.exe` в командной строке можно получить все ее опции. Например, заархивировать все содержимое подкаталога `/cat/` в файл `archcat.jar` можно так: `jar cfv archcat.jar cat`. Здесь опции компилятора `cfv` означают следующее: `c` - `create` создать новый архив, `f` - `file` указать имя файла, который и будет содержать заархивированный проект, `v` - `verbose`, выводить сообщения о процессе архивирования. Если нужно создать сжатый файл, содержащий программу на Java (например, файл `Hello.class`) и набор изображений, хранящихся в подкаталоге `/cat/` текущего каталога, то команда будет выглядеть так:

```
jar cfv archcat.jar Hello.class cat
```

Интерпретатор команд JVM способен извлекать нужные файлы из архива и загружать их. Для этого его нужно запускать с опцией `-cp` (`class path`). Например, если в командной строке текущим каталогом является каталог с программой `archcat.jar` и точка входа в программу (метод `main`) находится в классе `Hello`

```
java -cp archcat.jar Hello
```

Можно создать так называемый исполняемый jar-файл. Для этого в архиве должен присутствовать (текстовый) файл манифеста MANIFEST.MF с описанием содержимого архива. Для того, чтобы архив был исполняемым в файле манифеста должна быть запись, указывающая класс с методом main. Можно создать тестовый файл, например, mainif.txt который потом будет использован при создании файла манифеста со следующим содержимым:

Main-Class: HelloFrame

При вызове утилиты jar нужно задать опцию m - manifest

```
jar cmfv manif.txt HelloFrame.jar *.class
```

Теперь запускать программу на выполнение можно командой

```
java -jar HelloFrame.jar
```

ЗАДАНИЕ 4. 1) Заархивируйте файлы с расширением class, полученные после компиляции программы HelloFrame.java из предыдущего задания.

2) Запустите на выполнение программу HelloFrame из архива.

3) Создайте исполняемый архив для программы HelloFrame. Проверьте работоспособность программы.

IDE могут создавать исполняемый архив с Вашей программой. Например, в IntelliJ Idea для этого нужно перейти в File - Project Structure ... или нажать Ctrl+Alt+Shift+S. Выбрать Projects Settings - Artifacts, задать имя архивного файла. ОК. Затем Build - Build Artifacts ... Если в операционной системе связать расширение файла jar с командой вызова интерпретатора JVM (например, в Windows так java -jar %1), то программу из архива можно запускать как исполняемый файл, например, дважды щелкнув по его названию в каком-нибудь файловом менеджере. Описание утилиты jar в Java 15 доступно по ссылке <https://docs.oracle.com/en/java/javase/15/docs/specs/man/jar.html>.

5. Создание собственной документации.

Для создания собственной документации имеющей стандартный для Java вид набора взаимосвязанных страниц SDK предлагает программу javadoc.exe. Эта программа обрабатывает исходный java-файл и ищет комментарии специального вида: начинающиеся лексемой `/**` и заканчивающиеся `*/`. В этих комментариях можно располагать специальные тэги - атрибуты, начинающиеся с символа `@`. Например, тэг `@author` предназначен для указания автора программы. В файле `HelloFrame.java` Вы можете видеть несколько тэгов документирования Java.

ЗАДАНИЕ 5. Создайте документацию для `HelloFrame.java`. Для этого скопируйте сам этот файл в отдельный каталог и запустите javadoc:

```
javadoc HelloFrame.java
```

Посмотрите на html-файлы, созданные этой программой. Можно запустить, например, `index.html`. Отметьте, насколько форма подачи результата похожа на то, что содержится в стандартной документации.

Скачать руководство пользователя по javadoc (SDK 15) можно по ссылке <https://docs.oracle.com/en/java/javase/15/javadoc/javadoc-guide.pdf>.

Литература

1. <https://www.oracle.com/java/technologies/javase-downloads.html>
2. <https://jdk.java.net>
3. <https://adoptopenjdk.net/index.html>
4. <https://docs.oracle.com/javase/tutorial/>
5. <https://openjfx.io>
6. <https://docs.oracle.com/en/java/javase/15/jshell/java-shell-user-guide.pdf>
7. <https://code.visualstudio.com/docs/java/java-tutorial>

8. <https://www.jetbrains.com/idea/>
9. <https://netbeans.apache.org/download/index.html>
10. <https://www.eclipse.org/downloads/>
11. <https://docs.oracle.com/en/java/javase/15/docs/specs/man/jar.html>
12. <https://docs.oracle.com/en/java/javase/15/javadoc/javadoc-guide.pdf>

Редактор А.А. Литвинова

| | | |
|--------------------------------|---------|------------------|
| ЛР № 04779 от 18.05.01. | В набор | В печать |
| Объем 0,5 усл.п.л., уч.-изд.л. | Офсет. | Формат 60x84/16. |
| Бумага тип №3. | Заказ № | Тираж 120. Цена |

Издательский центр ДГТУ

Адрес университета и полиграфического предприятия:

344010, г. Ростов-на-Дону, пл. Гагарина, 1.