МИНИСТЕРСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «БЕЛОРУССКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ СВЯЗИ»

ОБОСОБЛЕННОЕ ПОДРАЗДЕЛЕНИЕ

«КОЛЛЕДЖ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ»

Отчет по лабораторной работе №1

«ЭТАПЫ РАБОТЫ С ПРОГРАММОЙ НА JAVA»

Выполнил учащийся гр. ТО-011

Ясюкевич Полина

Проверили

Янович Н.И.

Рогалевич П.И.

Минск, 2022

**Практические задания и методические указания**

1. Изучить порядок установки среды программирования на Java. теоретические сведения по теме: «Изучение этапов работы с программой на java в системе программирования».

2. Установить среду программирования и выполнить запуск тестовой программы согласно указаниям в теоретических сведениях.

3. Выполнить индивидуальные задания согласно варианту.

4. Создать пакеты и JAR-файлы согласно следующего описания (рисунок 2.1.1)

Задание 2

Пример 2.1

Код:

package com.company;

public class Main {

public static void main(String[] args) {

System.out.println("Hello, world!");

}

}

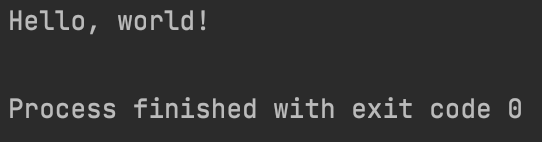


Рисунок 1 – Результат выполнения примера 2.1

Пример 2.2

Код:

package com.company;

public class Main {

public static void main(String[] args) {

int number1 = 11;

int number2 = 22;

int sum;

sum = number1 + number2;

System.out.print("The sum is "+sum);

}

}

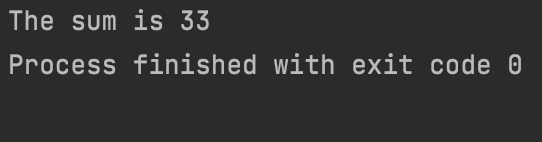


Рисунок 2 – Результат выполнения примера 2.2

Пример 2.3

Код:

package com.company;

public class Main {

public static void main(String[] args) {

System.out.printf("Hello%2d and %6s", 8, "HI!!!%n");

System.out.printf("Hi,%s%4d%n", "Hello", 88);

System.out.printf("Hi, %d %4.2f%n", 8, 5.556);

System.out.printf("Hi,%-4s&%6.2f%n", "Hi", 5.5);

System.out.printf("Hi, Hi, %.4f%n", 5.56);

}

}

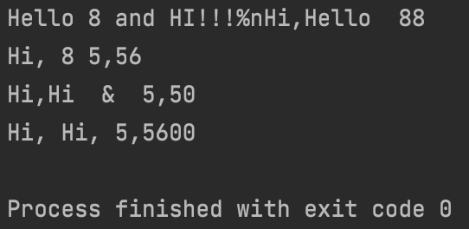


Рисунок 3 – Результат выполнения примера 2.3

Пример 2.4

Код:

package com.company;

public class Main {

public static void main(String[] args) {

java.util.Scanner in = new java.util.Scanner(System.in);

System.out.print("Enter a integer: ");

int number = in.nextInt();

System.out.println("You have entered: " + number);

}

}

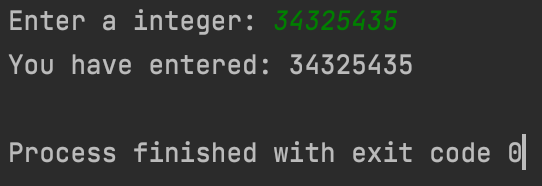


Рисунок 4 – Результат выполнения примера 2.4

Пример 2.5

Код:

package com.company;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("Enter a integer: ");

int number = in.nextInt();

System.out.println("You have entered: " + number);

}

}

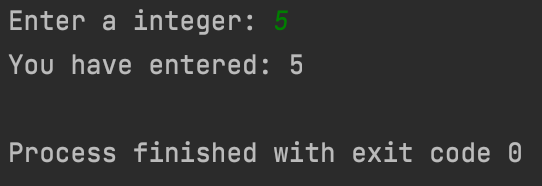


Рисунок 5 – Результат выполнения примера 2.5

Пример 2.6

Код:

package com.company;

import java.util.Scanner;

import java.awt.Graphics;

import java.util.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("Enter a integer: ");

int number = in.nextInt();

System.out.println("You have entered: " + number);

}

}

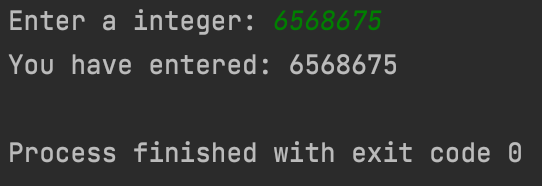


Рисунок 6 – Результат выполнения примера 2.6

Пример 2.7

Код:

package com.company;

import static java.lang.System.out;

import static java.lang.Math.\*;

public class Main {

public static void main(String[] args) {

out.println("Hello, PI is " + PI);

out.println("Square root of PI is " + sqrt(PI));

}

}

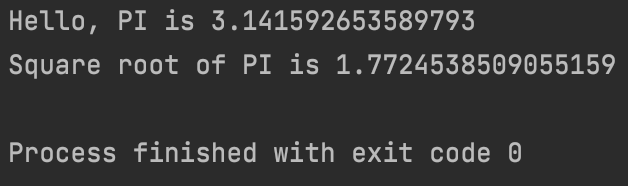


Рисунок 7 – Результат выполнения примера 2.7

Пример 2.8

Код:

public class GameMain {  
 public static final int *ROWS* = 3;  
 public static final int *COLS* = 3;  
}

public class GamePanel {  
 public static void main(String[] args) {  
  
 int[][] score = new int[GameMain.*ROWS*][GameMain.*COLS*];  
 for (int row = 0; row < GameMain.*ROWS*; ++row) {  
 for (int col = 0; col < GameMain.*COLS*; ++col) {  
 System.*out*.println(score[row][col]);  
 }  
 }  
 }  
}

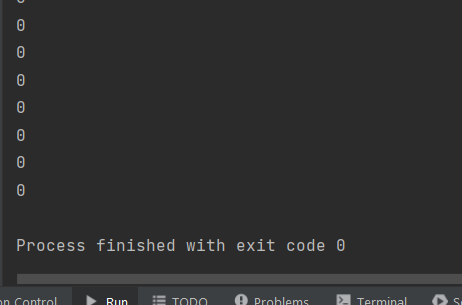


Рисунок 8 – Результат выполнения примера 2.8

Пример 2.9

Код:

package com.zzz.test;

public class HelloPackage

{

public static void main(String[] args)

{ System.out.println("Hello from a package...");} }

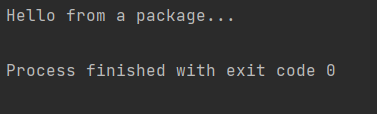


Рисунок 9 – Результат выполнения примера 2.9

Задание 3

Написать на языке Java программы для решения следующих задач.

10. Дана цена товара в рублях. Выразить ее в долларах США и евро.

import java.util.Scanner;

class Main

{

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("Введите цену в рублях: ");

int rub = in.nextInt();

double usd = (rub/2.53);

System.out.println("USD = "+usd);

double euro = (rub/2.51);

System.out.println("EURO = "+usd);

}

}

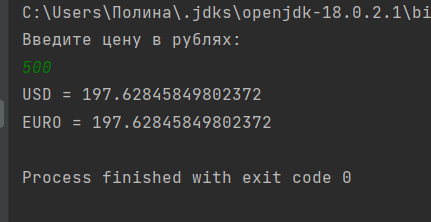


Рисунок 10 – Результат выполнения индивидуального задания 1

Задание 4

Создать пакеты и JAR-файлы согласно следующего описания (рисунок 11)

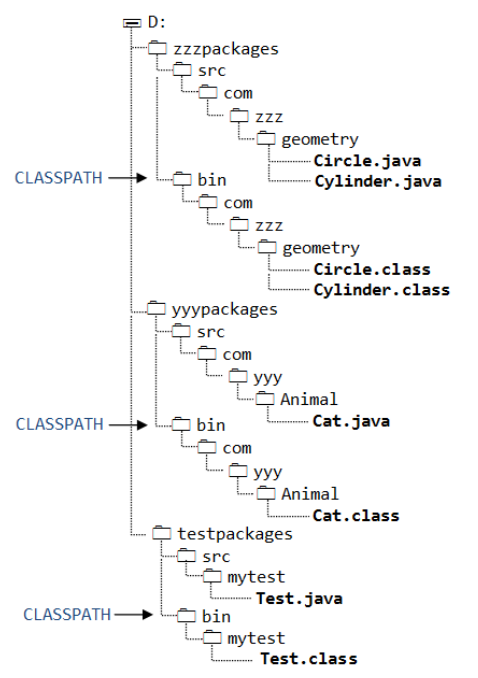


Рисунок 11

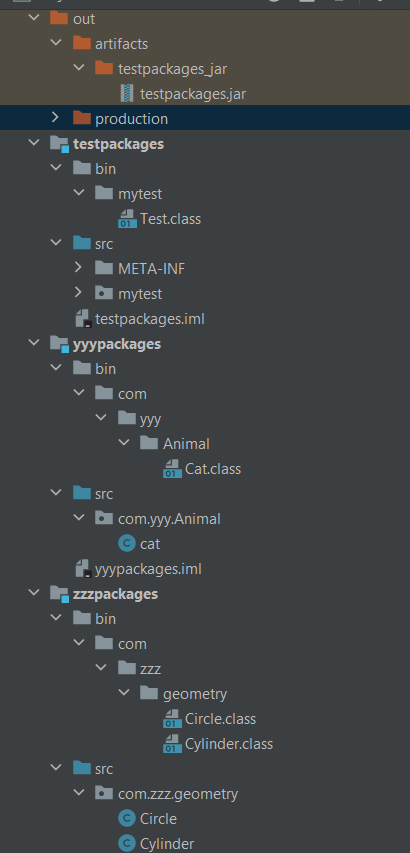


Рисунок 12 – Реализация структуры задания в IDE

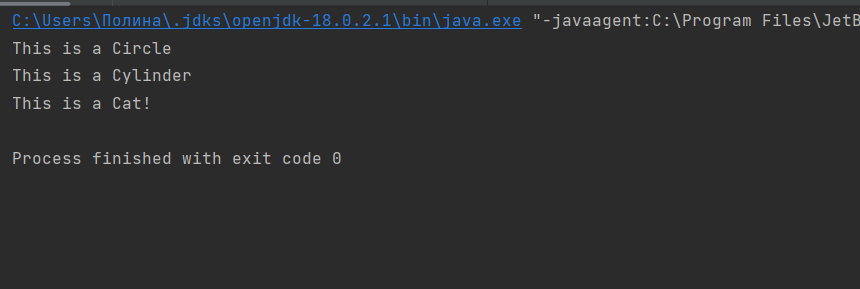


Рисунок 13 – Запуск программы из IDE

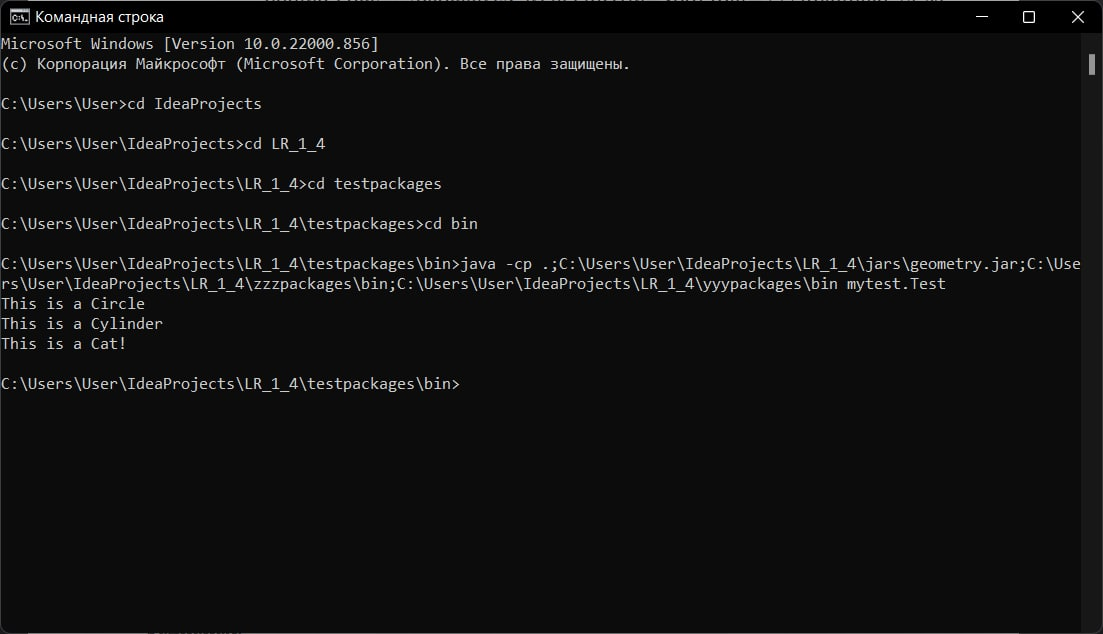


Рисунок 14 – запуск программы с помощь jar файла

**Контрольные вопросы**

**1. Приведите структуру программы на языке Java.**

Программа на языке Java представляет собой набор классов. В простейшем случае программа состоит из единственного класса, который обязательно содержит метод main(). Вот пример кода программы, которая выводит на экран монитора (консольный режим) строку «Hello, World!».

public class MyHello {  
public static void main(String[] args) {  
System.out.println("Hello, world!");  
}  
}

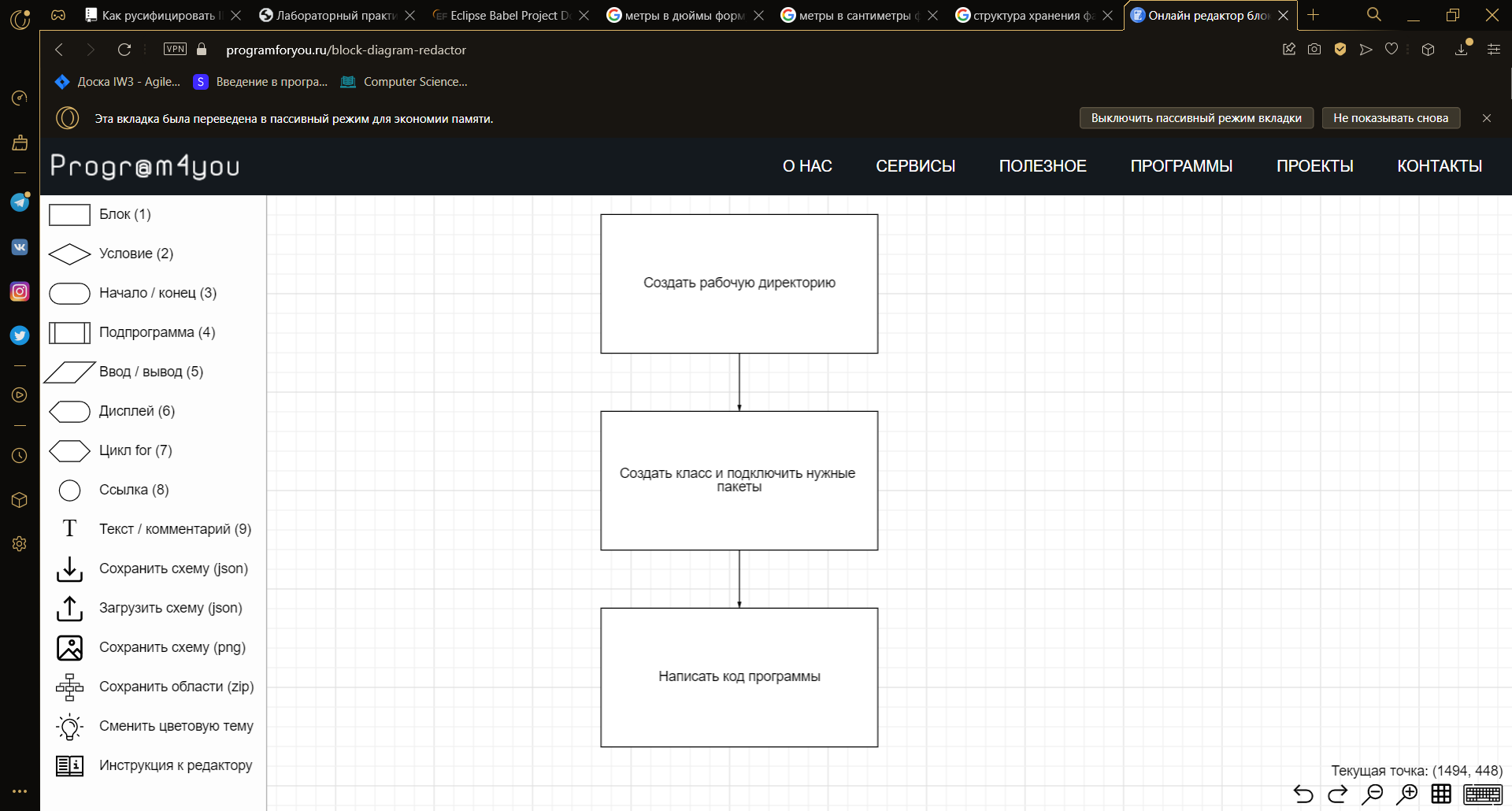
**2. Под каким именем необходимо сохранять файл с исходным кодом на языке Java?**

После написания этого кода, файл нужно сохранить под именем HelloWorld. java.

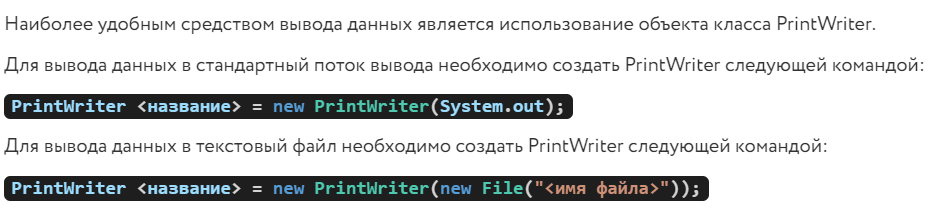
**3. Назовите компоненты среды программирования на Java?**

Компоненты среды программирования:  
1. Редактор  
2. Компилятор  
3. Компоновщик или редактор связей  
4. Загрузчик  
5. Отладчик  
6. Средства тестирования  
7. Интерпретатор

**4. Изобразите в виде блок-схемы алгоритм разработки и запуска программы на языке Java.**



**5. Назовите средства вывода данных в языке Java.**



**6. Что такое пакеты в Java? Какие опции и для чего используются при компиляции файлов в пакете?**

Пакет - это контейнер классов, который используются для обеспечения независимости пространства имен и ограничения доступа к классам.

По умолчанию java уже имеет ряд встроенных пакетов, например, java.lang, java.util, [java.io](https://vk.com/away.php?to=http%3A%2F%2Fjava.io&cc_key=) и т.д. Кроме того, пакеты могут иметь вложенные пакеты.

**7. Каково назначение операторов import и package?**

В Java пакеты (package) используются для предотвращения конфликтов с названиями, для контроля доступа, для облегчения поиска/нахождения и использования классов, интерфейсов, перечислений и аннотаций и т.д.

package com.company;

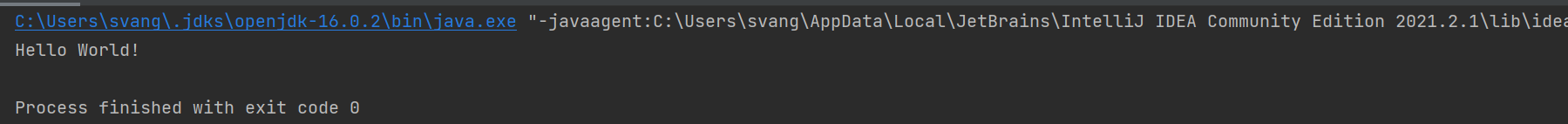
public class Main {

public static void main(String[] args)

{

System.out.println("Hello World!");

} }



**Оператор import**  
После оператора package, но до любого определения классов в исходном Java-файле, может присутствовать список операторов import. Пакеты являются хорошим механизмом для отделения классов друг от друга, поэтому все встроенные в Java классы хранятся в пакетах. Общая форма оператора import такова:  
  
import пакет1 [.пакет2].(имякласса|\*);

**8. Что представляют собой, для чего и как используются JAR-файлы в Java?**

JAR-файл представляет собой обычный ZIP-файл с некоторыми дополнениями. Основная задача этого архива — хранить файлы с классами.

**9. Что делать, если имена классов совпадают?**

Это происходит из-за того, что неявный вид строки, также известный как символ новой строки \n, добавляется ко всем пользовательским вводам терминала, поскольку он сообщает потоку начать новую строку. Вы можете смело учитывать это, используя [std::getline](https://legkovopros.ru/go.php?url=http%3A%2F%2Fen.cppreference.com%2Fw%2Fcpp%2Fstring%2Fbasic_string%2Fgetline" \t "_parent)при проверке нескольких строк ввода пользователя. Поведение по умолчанию std::getline будет читать все до и включая символ новой строки \n из объекта входного потока, который в этом случае является std::cin.

**10. Где JVM ищет классы?**

Программа запуска Java, **java**, запускает виртуальную машину Java. Виртуальная машина выполняет поиск и загрузку классов в таком порядке:

* **Классы начальной загрузки** - Классы, которые составляют платформу Java, включая классы в rt.jarи несколько других важных файлов jar.
* **Классы расширений** - Классы, использующие механизм расширения Java. Они упакованы в виде .jarфайлов, расположенных в каталоге расширений.
* **Классы пользователей** - Классы, определенные разработчиками и третьими сторонами, которые не используют преимущества механизма расширения. Вы определяете местоположение этих классов с помощью параметра-classpath в командной строке (предпочтительный метод) или с помощью переменной среды CLASSPATH. (См**. раздел Настройка пути к классам** для [Windows](https://docs.oracle.com/javase/1.5.0/docs/tooldocs/windows/classpath.html) или [Solaris](https://docs.oracle.com/javase/1.5.0/docs/tooldocs/solaris/classpath.html).)

По сути, эти три пути поиска объединяются, образуя простой путь к классу. Это похоже на ранее используемый путь к классу "плоский", но текущая модель имеет некоторые важные отличия:

* Относительно сложно случайно "скрыть" или опустить классы начальной загрузки.
* В общем случае вам нужно только указать расположение пользовательских классов. Классы начальной загрузки и классы расширений находятся "автоматически".
* Классы инструментов теперь находятся в отдельном архиве (tools.jar) и может использоваться только в том случае, если он включен в путь к пользовательскому классу (будет объяснено вкратце).

**11. Каковы структуры хранения файлов в средах Eclipse и NetBeans?**

Ваше предположение, что структура maven по умолчанию - src/target/pom.xml, неверна.  
Структура папок по умолчанию для четырех ваших исходных файлов - src/main/java, а для файлов ресурсов - src/main/resources. Если у вас есть статическое содержимое, такое как файлы html и javascript, они помещаются в папку src/main/webapp.

Ваша среда IDE (будь то Eclipse или Netbeans) имеет (и не должна) **НЕТ**управлять отображением структуры папок при импорте / создании проекта maven в среде IDE. Скорее это контролируется Maven.

В среде IDE просто отобразится структура папок в том виде, в котором она присутствует в данный момент.

**12. Приведите отличия в разработке программ на Java и C++.**

Разный синтаксис написания программ:

1. На С++ мы подключаем библиотеки, а на Java – пакеты.
2. На С++ для вывода на консоль используется *cout*, а на Java – куда более сложные команды.
3. На С++ для ввода с консоли используется *cin*, а на Java - in.nextInt().
4. Работа с Java строится на основе классов, которые в случае разработки на С++ имеют совершенно другую структуру.
5. В программах на Java нет чёткого указания на завершение программы.