Вопросы к защите лабораторной работы:

1. Java — язык программирования, который относится к объектно-ориентированным языкам(объекты взаимодействуют друг с другом) программирования, к языкам с сильной типизацией(не могут сложить строку и число)Создатели реализовали принцип WORA: «пиши один раз, запускай везде». Это значит, что написанное на Java приложение можно запустить на любой платформе, если на ней установлена среда исполнения Java. (Эта задача решается благодаря компиляции написанного на Java кода в байт-код. Этот формат исполняет JVM или виртуальная машина Java. JVM — часть среды исполнения Java (JRE). Виртуальная машина не зависит от платформы.) В Java реализован механизм управления памятью, который называется сборщиком мусора. Разработчик создаёт объекты, а JRE с помощью сборщика мусора очищает память, когда объекты перестают использоваться.Особенности: чувствительность к регистру — идентификаторы User и user в Java представляют собой разные сущности; для именования методов используется lowerCamelCase. Если название метода состоит из одного слова, оно должно начинаться со строчной буквы. Пример: firstMethodName(); для именования классов используется UpperCamelCase. Если название состоит из одного слова, оно должно начинаться с прописной буквы. Пример: FirstClassName. название файлов программы должно точно совпадать с названием класса с учётом чувствительности к регистру. Например, если класс называется FirstClassName, файл должен называться FirstClassName.java; идентификаторы всегда начинаются с буквы (A-Z, a-z), знака $ или нижнего подчёркивания \_; Ключевой особенностью языка Java является то, что его код сначала транслируется в специальный байт-код, независимый от платформы. А затем этот байт-код выполняется виртуальной машиной JVM (Java Virtual Machine).

2. JDK нужен для разработки (это компилятор, отладчик и т.д.).JRE нужен для запуска Java программ (содержит в себе JVM).JDK и JRE содержат JVM, которая нужна для запуска программ на Java.JVM является сердцем языка программирования Java и обеспечивает независимость от платформы. Java-модель памяти, используемая внутри JVM, делит память на стеки потоков (thread stacks) и кучу

Java Development Kit является основным компонентом среды Java и предоставляет все инструменты, исполняемые и бинарные файлы, которые нужны для компиляции, отладки и выполнения программы на Java. JDK является платформо-зависимым программным обеспечением, поэтому есть отдельные инсталляторы для Windows, Mac и Unix-систем. Можно сказать, что JDK является надстройкой JRE, так как он содержит JRE с Java-компилятором, отладчиком и базовыми классами. Текущая версия JDK 1.8 также известна как Java 8.

JRE является реализацией JVM, которая предоставляет платформу для выполнения Java-программ. JRE состоит из виртуальной машины Java, бинарных файлов и других классов. JRE не содержит инструменты для разработки (компилятор Java, отладчик и т.д). Если вы хотите запустить любую Java программу, вы должны установить JRE.

3. Java byte(8 бит)Short(16 бит)Int(31 бит= 4 байт)Long(64 бит = 8 байт)Java float и Java double(добл точнее(64 бит)флоат(32) Данные типы называются типами с плавающей точкой. То есть это не целочисленные типы.Java boolean(Следующий тип – булевский (логический тип). Он может принимать значения только true или false, которые являются ключевыми словами.)Java char(16 бит хранит символы в кодировке Unicode) Основные особенности стека:Он заполняется и освобождается по мере вызова и завершения новых методовПеременные в стеке существуют до тех пор, пока выполняется метод в котором они были созданыЕсли память стека будет заполнена, Java бросит исключение java.lang.StackOverFlowErrorДоступ к этой области памяти осуществляется быстрее, чем к кучеЯвляется потокобезопасным, поскольку для каждого потока создается свой отдельный стек. Особенности кучи: Помимо рассмотренных ранее, куча имеет следующие ключевые особенности:  
Когда эта область памяти полностью заполняется, Java бросает java.lang.OutOfMemoryErroДоступ к ней медленнее, чем к стекуЭта память, в отличие от стека, автоматически не освобождается. Для сбора неиспользуемых объектов используется сборщик мусора В отличие от стека, куча не является потокобезопасной и ее необходимо контролировать, правильно синхронизируя код.

4. Декларация - это когда вы объявляете переменную с именем, а переменная может быть объявлена ​​только один раз. Инициализация переменной – присвоение переменной значения.

5. Ветвление — это алгоритмическая конструкция, в которой в зависимости от истинности некоторого условия, выполняется одна из нескольких последовательностей действий.Операторы ветвления Синтаксис оператора ветвления if else. Многовариантное ветвление, оператор switch . Конструкция if else может оказаться неудобной, если вы стоите перед необходимостью сделать выбор из нескольких вариантов. Оператор switch обеспечивает ясный способ переключения между различными частями программного кода в зависимости от значения одной переменной или выражения

Цикл — это разновидность управляющей конструкции, предназначенная для организации многократного исполнения набора инструкцийВ основе ветвления и отдельных циклов лежат логические операторы сравнения, определяющие необходимость выполнение следующих строк кода или перехода к другим.. Циклы javaВ java существует два типа циклов:цикл типа «пока» - while и do…while , типа «n-раз» - for (инициализация; условие; итерация)

6. Все операторы Java можно разделить на следующие группы:

* арифметические операторы;
* операторы сравнения;
* побитовые операторы;& (и) | (или) ^(лог или) ~ (допол) биты
* логические операторы;( && и) (|| или) (! Не)
* операторы присваивания;
* прочие операторы.( Тернарный оператор – оператор, который состоит из трех операндов и используется для оценки выражений типа boolean.)
* Оператор instanceof – проверяет, является ли объект определенного типа (типа класса или типа интерфейса) и используется только для переменных ссылочного объекта. Оператор instanceof записывается в виде:

7. Вычислять значения по модулю;

* Вычислять значения тригонометрических функций;
* Возводить числа в степень;
* Извлекать квадратный и кубический корни;
* Генерировать случайные числа;
* Округлять числа.

8. Общий вид инструкции форматирования следующий:

%[argument\_index$][flags][width][.precision]conversion

* % — специальный символ, обозначающий начало инструкций форматирования.
* [argument\_index$] — целое десятичное число, указывающее позицию аргумента в списке аргументов. Ссылка на первый аргумент "1$", ссылка на второй аргумент "2$" и т.д. Не является обязательной частью инструкции. Если позиция не задана, то аргументы должны находиться в том же порядке, что и ссылки на них в строке форматирования.
* [flags] — специальные символы для форматирования. Например, флаг "+" означает, что числовое значение должно включать знак +, флаг "-" означает выравнивание результата по левому краю, флаг «,» устанавливает разделитель тысяч у целых чисел. Не является обязательной частью инструкции.
* [width] — положительное целое десятичное число, которое определяет минимальное количество символов, которые будут выведены.  Не является обязательной частью инструкции.
* [.precision] — не отрицательное целое десятичное число с точкойперед ним. Обычно используется для ограничения количества символов. Специфика поведения зависит от вида преобразования. Не является обязательной частью инструкции.
* conversion — это символ, указывающий, как аргумент должен быть отформатирован. Например d для целых чисел, s для строк, f для чисел с плавающей точкой. Является обязательной частью инструкции.

Класс — это, по сути, шаблон для объекта. Он определяет, как объект будет выглядеть и какими функциями обладать. **Каждый объект является объектом какого-то класса**. **Методы** — это функционал твоего класса; то, что объекты этого класса умеют делать. С одним из методов ты уже знаком — это метод main(). Но метод main, как ты помнишь, является статическим — то есть он принадлежит всему классу (логика такая же, как с переменными).

[Java Development Kit](https://docs.oracle.com/en/java/javase/13/docs/specs/man/index.html) — комплект разработчика приложений на языке Java. Он включает в себя [Java Development Tools](https://docs.oracle.com/javase/6/docs/technotes/tools/index.html) и среду выполнения Java — [JRE](https://www.oracle.com/technetwork/java/javase/jre-8-readme-2095710.html) (Java Runtime Environment).  
  
**Java development tools** включают в себя около 40 различных тулов: javac (компилятор), java (лаунчер для приложений), javap (java class file disassembler), jdb (java debugger) и др.  
  
Среда выполнения JRE — это пакет всего необходимого для запуска скомпилированной Java-программы. Включает в себя виртуальную машину [JVM](https://docs.oracle.com/javase/6/docs/technotes/guides/vm/index.html) и библиотеку классов Java — **Java Class Library**.  
  
**JVM** — это программа, предназначенная для выполнения байт-кода. Первое преимущество JVM — это принцип *“Write once, run anywhere”*. Он означает, что приложение, написанное на Java, будет работать одинаково на всех платформах. Это является большим преимуществом JVM и самой Java.

2. Выполнение кода на JVM

Согласно [спецификации Java SE](https://docs.oracle.com/javase/specs/), для того, чтобы получить код, работающий в JVM, необходимо выполнить 3 этапа:

* Загрузка байт-кода и создание экземпляра класса Class  
  Грубо говоря, чтобы попасть на JVM, класс должен быть загружен. Для этого существуют отдельные класс-загрузчики, к ним мы вернемся чуть позже.
* Связывание или линковка  
  После загрузки класса начинается процесс линковки, на котором байт-код разбирается и проверяется. Процесс линковки в свою очередь происходит в 3 шага:  
    
  — verification или проверка байт-кода: проверяется корректность инструкций, возможность переполнения стека на данном участке кода, совместимость типов переменных; проверка происходит один раз для каждого класса;  
  — preparation или подготовка: на данном этапе в соответствии со спецификацией выделяется память под статические поля и происходит их инициализация;  
  — resolution или разрешение: разрешение символьных ссылок (когда в байт-коде мы открываем файлы с расширением .class, мы видим числовые значения вместо символьных ссылок).
* Инициализация полученного объекта Class  
  На последнем этапе класс, который мы создали, инициализируется, и JVM может начинать его исполнение.

**Стек** — это область памяти в Java, которая работает по схеме LIFO — “*Last in — Fisrt Out*” или “*Последним вошел, первым вышел*”.Он нужен для того, чтобы хранить методы. Переменные в стеке существуют до тех пор, пока выполняется метод в котором они были созданы.

Ещё одна область памяти в Java — **Heap** или **куча**. Она используется для хранения объектов и классов. Новые объекты всегда создаются в куче, а ссылки на них хранятся в стеке. Все объекты в куче имеют глобальный доступ, то есть к ним можно получить доступ из любой точки приложения.

*main* – имя метода, это идентификатор, который JVM ищет при выполнении программы Java. является, как правило, точкой входа в программу.

Ключевое слово static означает, что метод является статическим, т. е. не требует для своего вызова наличие экземпляра класса. Ключевое слово void означает, что метод не возвращает никакого значения. Все эти три слова обязательно должны присутствовать в описании метода.

Запись (String[] args) после имени метода представляет собой список его параметров. Имеется лишь один параметр args, содержащий ссылку на массив строк. Этот массив формируется из строки, переданной классу, содержащему метод main(), при запуске этого класса на выполнение командой java.

An OutOfMemoryError— это исключение, создаваемое виртуальной машиной Java (JVM), поскольку ей необходимо выделить память для (нового) объекта, но для объекта доступно недостаточно памяти. JVM сначала попытается освободить память, используемую мертвыми объектами, [запустив сборщик мусора](https://stackoverflow.com/questions/12298725/is-the-garbage-collector-guaranteed-to-run-before-out-of-memory-error).

JID - Java Image Downloader имеет понятный и интуитивно понятный макет, который позволяет настраивать выделенные параметры всего несколькими щелчками мыши. легкая утилита для Windows, предназначенная для того, чтобы помочь пользователям загружать изображения из Интернета и сохранять их на компьютере для просмотра в автономном режиме.

Приложение позволяет загружать изображения, указав действительный URL-адрес или импортировав список с URL-адресами из обычного текстового файла. Автоматически извлекать ссылки на изображения с форумов и т. Д.