
QA
מע

אלכס גורבצ'וב
АЛЕКСЕЙ ГОРБАЧЁВ



INTRODUCTION TO AUTOMATION

ВВЕДЕНИЕ В АВТОМАТИЗАЦИЮ

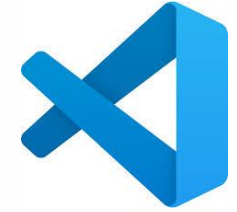


VISUAL STUDIO

VISUAL STUDIO



Visual Studio



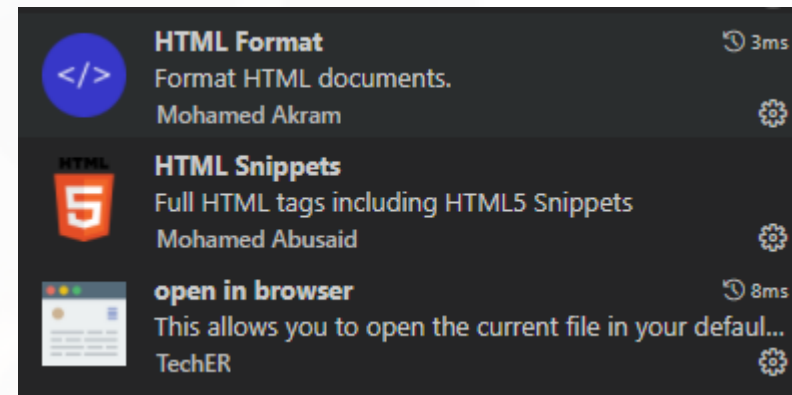
Visual Studio

Visual Studio Code - это упрощенный, но мощный редактор исходного кода, который работает на компьютере и доступен для Windows, macOS и Linux.

Имеет встроенную поддержку JavaScript, TypeScript и Node.js, а также обширную экосистему расширений для других языков и сред выполнения (например, C++, C#, Java, Python, PHP, Go, .NET).



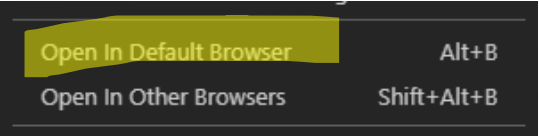
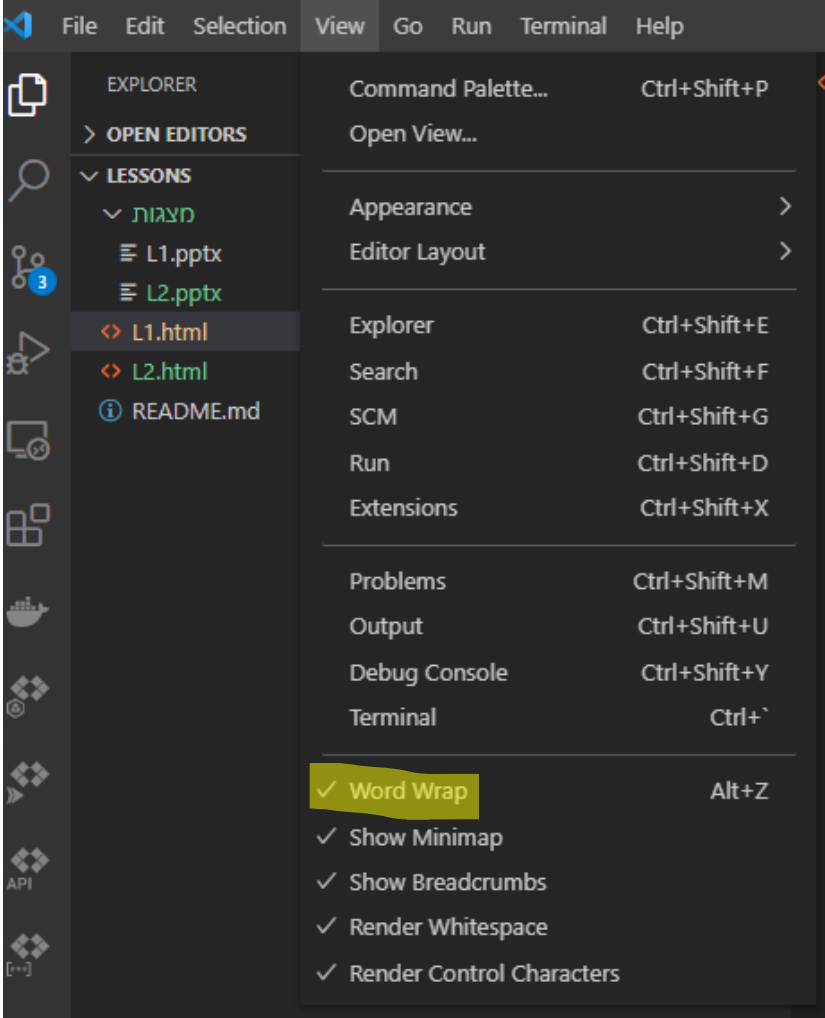
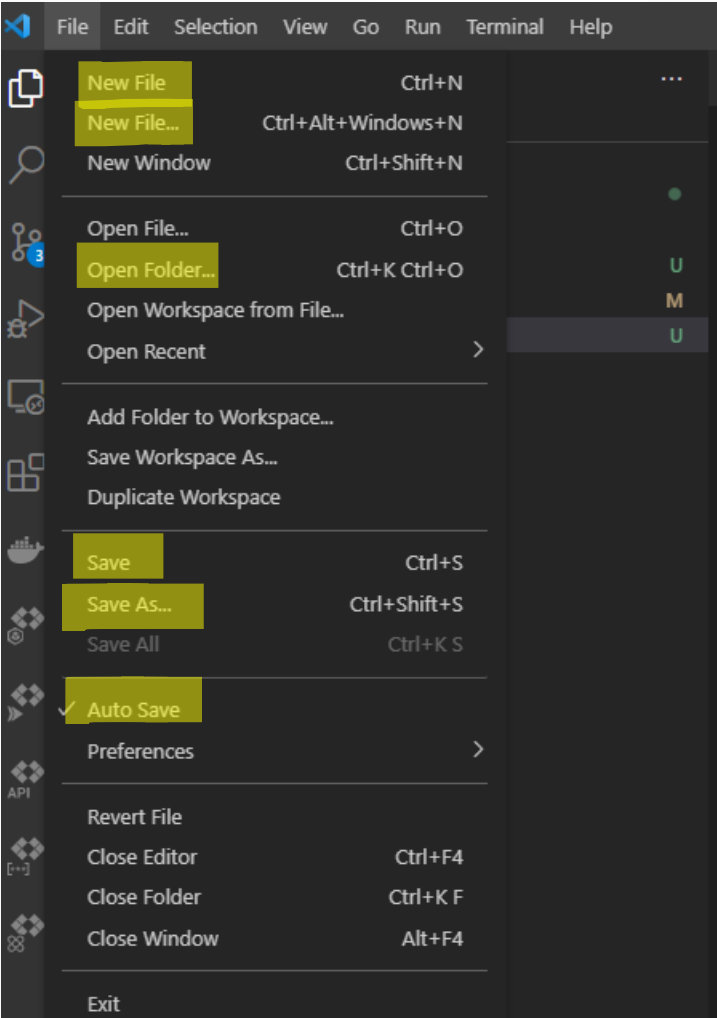
Visual Studio Code



Visual Studio



Visual Studio






GIT
GIT





Действия при пожаре:

1.  `git commit`
2.  `git push`
3.  Покинуть помещение

GitHub Desktop

GitHub Desktop - это приложение, которое позволяет вам взаимодействовать с GitHub с помощью графического интерфейса вместо командной строки или веб-браузера.

GitHub Desktop поощряет команду к совместной работе, используя передовые методы работы с Git и GitHub. Можно использовать GitHub Desktop для выполнения большинства команд Git с рабочего стола с визуальным подтверждением изменений.

Установка: [GitHub Desktop](#).

Git



FileEditViewRepositoryBranchHelp

Current repository
desktop

Current branch
esc-pr #3972 ✓

Fetch origin
Last fetched 2 minutes ago

ChangesHistory

Add event handler to dropdown component

iAmWillShepherd and Markus Olsson committed c79e71c 1 changed file

Co-Authored-By: Markus Olsson <niik@users.noreply.github.com>

app\src\ui\toolbar\dropdown.tsx

@@ -145,6 +145,10 @@ export class ToolbarDropdown extends React.Component<
145 145 this.state = { clientRect: null }
146 146 }
147 147
148 + private get isOpen() {
149 + return this.props.dropdownState === 'open'
150 + }
151 +
152 private dropdownIcon(state: DropdownState): OcticonSymbol {
153 // @TODO: Remake triangle octicon in a 12px version,
154 // right now it's scaled badly on normal dpi monitors.
@@ -249,6 +253,13 @@ export class ToolbarDropdown extends React.Component<
249 253 }
250 254 }
251 255
256 + private onFoldoutKeyDown = (event: React.KeyboardEvent<HTMLElement>) => {
257 + if (!event.defaultPrevented && this.isOpen && event.key === 'Escape') {
258 + event.preventDefault()

JAVA
JAVA



Как выглядит язык программирования?

Каждый язык программирования основывается на системе программирования.

Системой программирования будем называть комплекс программных средств, предназначенных для кодирования, тестирования и отладки программного обеспечения. Другими словами, это набор специализированных программных продуктов, которые являются инструментальными средствами разработчика.

Система программирования, как правило, включает следующие программные компоненты:

- редактор текста
- транслятор с соответствующего языка
- компоновщик (редактор связей)
- отладчик
- библиотеки подпрограмм

Как выглядит язык программирования?

Любая система программирования может работать только в соответствующей ОС, под которую она и создана, однако при этом она может позволять разрабатывать программное обеспечение и под другие ОС.

Редактор текста - это программа для ввода и модификации текста.

Программа, подготовленная на каком-либо языке программирования, называется исходным модулем и представляет собой текстовый файл с соответствующим расширением.

Трансляторы предназначены для преобразования программ, написанных на языках программирования, в программы на машинном языке. В качестве входной информации трансляторы применяют исходные модули и формируют в результате своей работы объектные модули, являющиеся входной информацией для редактора связей.

Как выглядит язык программирования?

Трансляторы делятся на два класса: компиляторы и интерпретаторы.

- Компилятор переводит весь исходный модуль на машинный язык. При компиляции одна и та же программа имеет несколько представлений - в виде текста и в виде выполняемого файла.
- Интерпретатор последовательно переводит на машинный язык каждый оператор исходного модуля и сразу же выполняет его. Основной недостаток интерпретатора - низкая скорость работы интерпретируемых программ (во время выполнения программы необходим перевод каждого оператора на машинный язык). Преимущество интерпретатора перед компилятором состоит в том, что программа пользователя имеет одно представление - в виде текста.

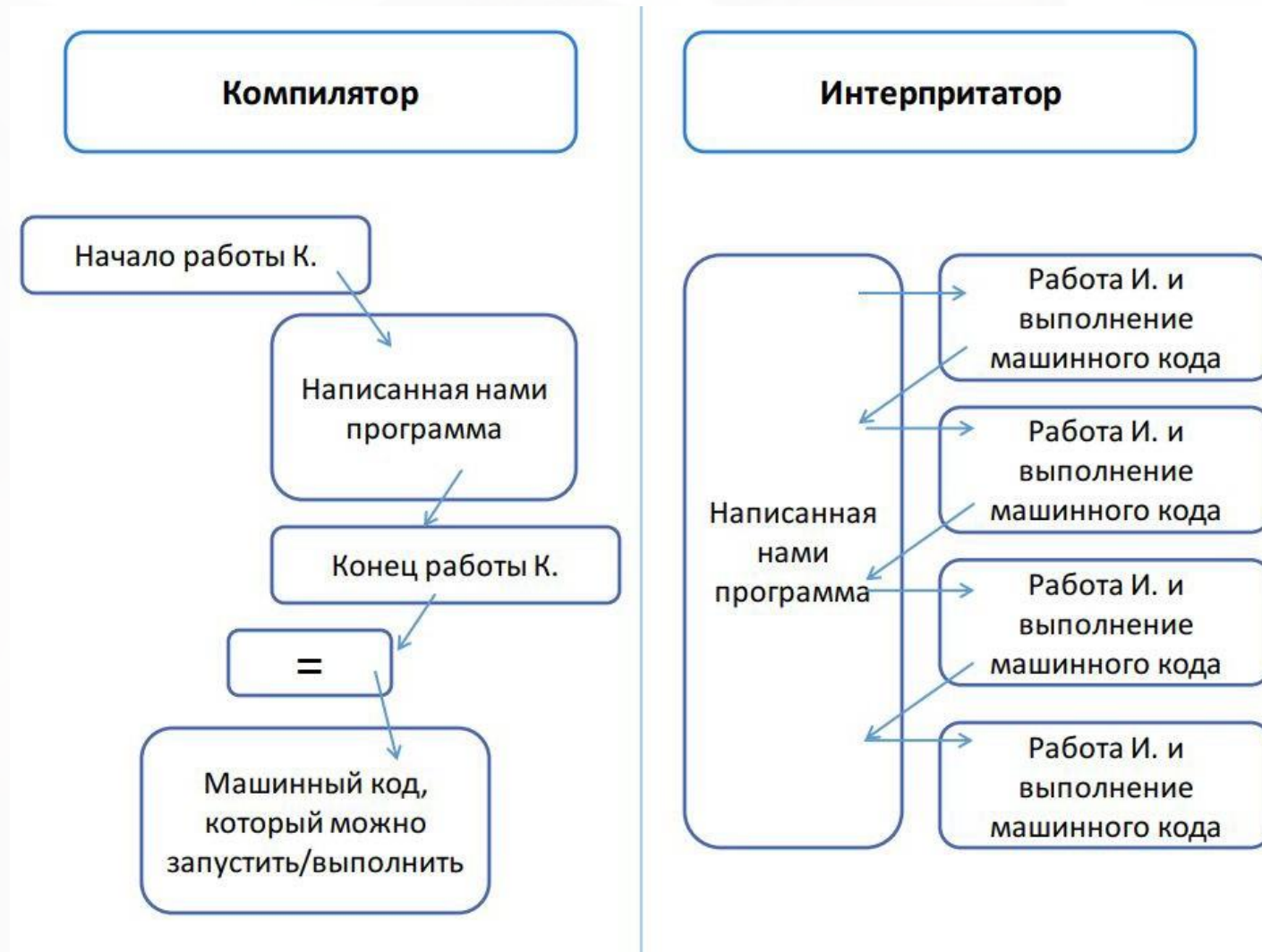
Как выглядит язык программирования?



Java



Как выглядит язык программирования?

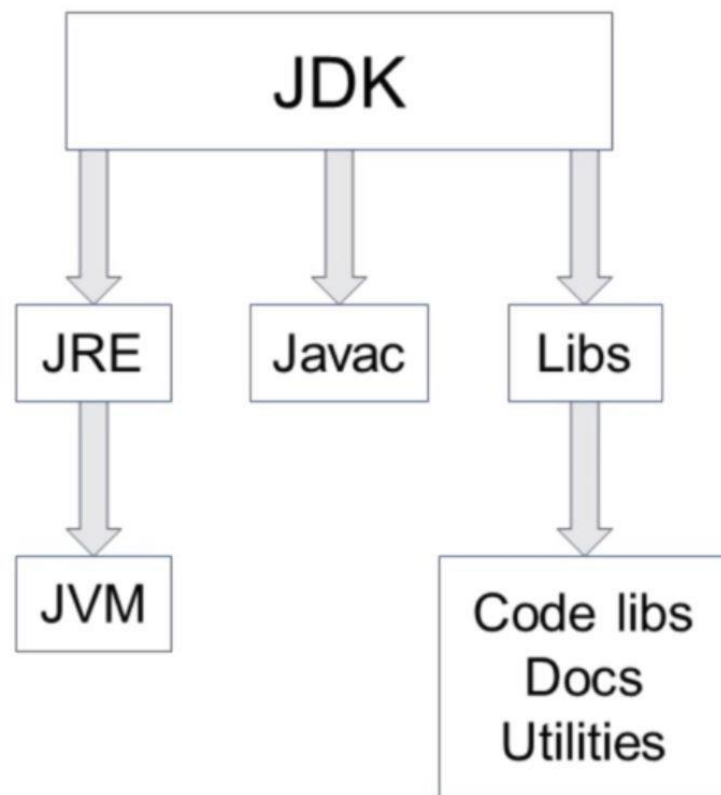


Java



Компоненты Java

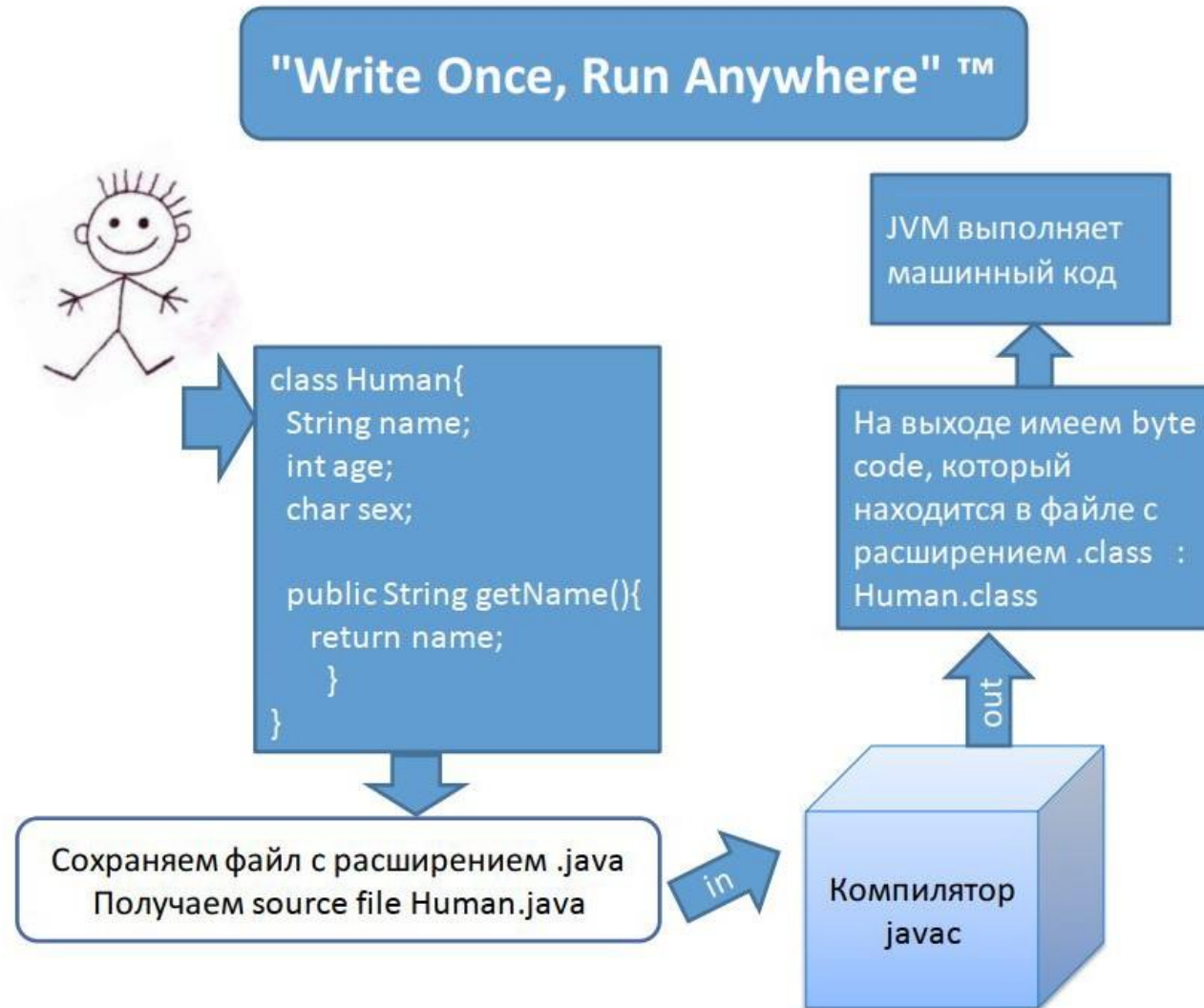
Компоненты Java



Java



Компоненты Java



Java



Установка

- Установим 17 версию языка, так как она самая популярная и является версией LTS, с поддержкой до 2029 года. Скачать можно [тут](#).
- После скачивания – устанавливаем как простую программу (пару кликов Next).
- Запустим Visual Code.
- Напишем код:

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

- Сохраним как HelloWorld.java
- Запустим компилятор 

Java



Первые шаги

- Выведите на консоль ваше имя, город проживания и возраст. Каждое в новой строке.
- Выведите на консоль следующую цитату:

```
It's all about quality of life  
and finding a happy balance between work  
and friends and family.
```

Что такое переменная в Java?

Переменная - это некоторый контейнер, в котором может храниться значение для дальнейшего использования в программе.

Помните, как в школе было: $y = x + 1$

И в зависимости от того, какие значения принимает переменная x , меняется значение переменной y .

Если $x = 1$, тогда $x + 1 = 2$

Если $x = 2$, тогда $x + 1 = 3$

В Java переменные играют такую же роль, как и в приведенном примере со школы $y = x + 1$. Они выполняют роль контейнера для разных значений, которые можно подставить в переменную. В приведённом примере - в переменную x .

Java



Типы переменных в Java. Объявление переменной

В Java можно указать, какие именно значения может принимать переменная.

Для этого все переменные сгруппировали в 4 группы:

- Целочисленные (к ним относятся byte, short, int, long)
- С плавающей точкой (к ним относятся float, double)
- Символы (char)
- Логические (boolean)

Типы переменных в Java. Объявление переменной

При создании переменной необходимо указать:

- Имя переменной (*идентификатор*)
- Тип переменной
- Начальное значение (*необязательно*)



Инициализация переменной – это первое присвоение ей значения.

Типы переменных в Java. Объявление переменной



Имена переменных должны быть понятны и передавать смысл хранимого значения.

```
int myVariable = 2;
```

Первый символ идентификатора

Допустимые символы	Недопустимые символы
A-Z, a-z, _ , \$	0-9, . , , , ? , : , * , > , < , = , & , # , ! и т.п.

Остальные символы идентификатора

Допустимые символы	Недопустимые символы
A-Z, a-z, _ , \$, 0-9	@ , . , , , ? , : , * , > , < , = , & , # , ! и т.п.

Типы переменных в Java. Объявление переменной

- 1) В идентификаторах допустимо использовать символы алфавита, нижнего подчеркивания и символ доллар:

`myVariable, my_Variable, _MyVariable, my$Variable, $myVariable`

- 2) Использование цифр недопустимо только на первой позиции:

`myVariable1, my1Variable, 1MyVariable`

- 3) Нельзя использовать в качестве идентификаторов зарезервированные ключевые слова:

~~`extends, false, import`~~, `intMyVar`

- 4) Язык JAVA чувствителен к регистру, поэтому если вы напишите их в разном регистре – это будут различные переменные:

`myVariable, MyVariable, myvariable`

Типы переменных в Java. Объявление переменной

Ключевые слова – это предварительно определенные зарезервированные идентификаторы, имеющие специальные значения для компилятора.

<code>abstract</code>	<code>assert</code>	<code>boolean</code>	<code>break</code>	<code>byte</code>
<code>case</code>	<code>catch</code>	<code>char</code>	<code>class</code>	<code>const</code>
<code>continue</code>	<code>default</code>	<code>do</code>	<code>double</code>	<code>else</code>
<code>enum</code>	<code>extends</code>	<code>final</code>	<code>finally</code>	<code>float</code>
<code>for</code>	<code>goto</code>	<code>if</code>	<code>implements</code>	<code>import</code>
<code>instanceof</code>	<code>int</code>	<code>interface</code>	<code>long</code>	<code>native</code>
<code>new</code>	<code>package</code>	<code>private</code>	<code>protected</code>	<code>public</code>
<code>return</code>	<code>short</code>	<code>static</code>	<code>strictfp</code>	<code>super</code>
<code>switch</code>	<code>synchronized</code>	<code>this</code>	<code>throw</code>	<code>throws</code>
<code>transient</code>	<code>try</code>	<code>void</code>	<code>volatile</code>	<code>while</code>

Типы переменных в Java. Объявление переменной

целочисленные

с плавающей точкой

символы

логические

ТИП ДАННЫХ	ДИАПАЗОН ДОПУСТИМЫХ ЗНАЧЕНИЙ	ОБЪЁМ ЗАНИМАЕМОЙ ПАМЯТИ
byte	от -128 до 127	1 байт
short	от -32768 до 32767	2 байта
int	от -2147483648 до 2147483647	4 байта
long	от -9223372036854775808 до 9223372036854775807	8 байт
float	от -3.4E+38 до 3.4E+38	4 байта
double	от -1.7E+308 до 1.7E+308	8 байт
char	от 0 до 65536	2 байта
boolean	true или false	Для хранения значения этого типа достаточно 1 бита, но в реальности память такими порциями не выделяется, поэтому переменные этого типа могут быть по-разному упакованы виртуальной машиной

Типы переменных в Java. Целочисленные: byte, short, int, long.

- byte может принимать значения от -128 до 127 и при этом занимает 1 байт памяти
- short принимает значения от -32768 до 32767 и занимает 2 байта памяти
- int от -2147483648 до 2147483647 и занимает 4 байта памяти
- long от -9223372036854775808 до 9223372036854775807 и занимает 8 байтов памяти

Чаще всего при написании программ на Java используется int. Это самый распространенный целочисленный тип. Более того, тип int в Java является типом "по умолчанию" для целочисленных типов.

Типы переменных в Java. Целочисленные: byte, short, int, long.

ТИП ДАННЫХ	ДИАПАЗОН ДОПУСТИМЫХ ЗНАЧЕНИЙ	ОБЪЁМ ЗАНИМАЕМОЙ ПАМЯТИ
byte	от -128 до 127	1 байт
short	от -32768 до 32767	2 байта
int	от -2147483648 до 2147483647	4 байта
long	от -9223372036854775808 до 9223372036854775807	8 байт

Java



Типы переменных в Java. С плавающей точкой: float, double.

ТИП ДАННЫХ	ДИАПАЗОН ДОПУСТИМЫХ ЗНАЧЕНИЙ	ОБЪЁМ ЗАНИМАЕМОЙ ПАМЯТИ
float	от $-3.4\text{E}+38$ до $3.4\text{E}+38$	4 байта
double	от $-1.7\text{E}+308$ до $1.7\text{E}+308$	8 байт

Типы переменных в Java. Символы: char.

В Java для char используется кодировка Unicode.

Диапазон допустимых значений char от 0 до 65536 и при этом занимает 2 байта памяти. Отрицательных значений не существует.

На самом деле переменная типа char хранит не сам символ, а его числовой код из таблички Unicode, по этому мы можем проводить целочисленные операции над символами.

ТИП ДАННЫХ	ДИАПАЗОН ДОПУСТИМЫХ ЗНАЧЕНИЙ	ОБЪЁМ ЗАНИМАЕМОЙ ПАМЯТИ
char	от 0 до 65536	2 байта

Типы переменных в Java. Логические: `boolean`.

Логические или их еще называют булевы значения могут принимать только одно из двух возможных значений: `true` или `false`.

В языке программирования Java, в отличие от разговорного, всё очень точно:

- либо `true` (с английского - "истина")
- либо `false` (с английского - "ложь")

ТИП ДАННЫХ	ДИАПАЗОН ДОПУСТИМЫХ ЗНАЧЕНИЙ	ОБЪЁМ ЗАНИМАЕМОЙ ПАМЯТИ
<code>boolean</code>	<code>true</code> или <code>false</code>	Для хранения значения этого типа достаточно 1 бита, но в реальности память такими порциями не выделяется, поэтому переменные этого типа могут быть по-разному упакованы виртуальной машиной

Java



Как присвоить значение переменной в Java?

Присваивание значения переменной осуществляется с помощью знака =

Допустим, мы хотим присвоить переменной k типа int значение 10. Это можно это сделать 2 способами:

Способ 1:

```
class Test {  
    public static void main(String args[]){  
        int k;  
        k = 10;  
        System.out.println (k);  
    }  
}
```


Java



Как присвоить значение переменной в Java?

Способ 2:

```
class Test {  
    public static void main(String args[]){  
        int k = 10;  
        System.out.println (k);  
    }  
}
```

Задание.

- Создайте по 2 переменные всех целочисленных типов данных в следующем виде:
 - 2 переменные типа byte
 - 2 переменные типа short
 - 2 переменные типа int
 - 2 переменные типа longИ вывести их на консоль.
- Создать по 2 переменные типов данных float, double и boolean. И вывести их на консоль.

Что такое инициализация переменной?

Инициализация - это присвоение начального значения переменной. То есть, если мы только создали переменную и не задали ей еще никакого значения, то данная переменная неинициализированна.

Что такое инициализация переменной?

```
class Test {  
    public static void main(String args[]){  
        byte number1 = 15;  
        short number2 = 100;  
        long number3 = 100000000;  
        String title = "Обожаю Java";  
        char letter = 'M';  
        double sum = 145.34567;  
        float pi = 3.14f;  
        boolean result = true;  
        System.out.println(number1);  
        System.out.println(number2);  
        System.out.println(number3);  
        System.out.println (title);  
        System.out.println (letter);  
        System.out.println (sum);  
        System.out.println (pi);  
        System.out.println (result);  
    }  
}
```

Как оставлять комментарии в коде?

Правильная практика программирования - прописывать комментарии.
Благодаря этому:

- Вы всегда легко вспомните что к чему в программе. Стандартная практика - месяц не подходил к коду, открываешь его и сам не веришь, что это ты написал... И вот в этот момент так тепло и радостно на душе, когда видишь комменты...
- И тот, кто после Вас будет работать с Вашим кодом, скажет Вам огромное спасибо. Вас же не будет под боком, чтоб спросить: "А что это ты, друг, имел ввиду...?«

Так что пишите комменты - позаботьтесь о себе и о последующих поколениях...

Как оставлять комментарии в коде?

- Если комментарий написан в 1 строчку, используем две косые черты `//` и далее пишем наш коммент

```
System.out.println (k); //выводит на консоль значение переменной k
```

- Если же комментарий у нас не в 1 строчку, а в 2 и более строк, то используем такую `/*` в начале и `*/` в конце

```
/*
```

```
Программа, демонстрирующая вывод значения  
переменной на консоль
```

```
*/
```

Как выводить сообщения в консоль?

Для вывода сообщений в консоль существуют 2 метода: **System.out.println ()** и **System.out.print ()**. В чём же отличие?

Отличие между **System.out.println ()** и **System.out.print ()** в том что:

- **System.out.println ()** выводит сообщение на экран и после этого осуществляется перевод курсора на новую строку
- **System.out.print ()** выводит сообщение на экран и после этого не осуществляется перевод курсора на новую строку

Java



Как выводить сообщения в консоль?

```
class Test {  
    public static void main(String[] args) {  
        System.out.println("Я изучаю");  
        System.out.println ("Java");  
    }  
}
```

```
class Test {  
    public static void main(String[] args) {  
        System.out.println("Я изучаю");  
        System.out.println ("Java");  
        System.out.println ("Ура!");  
    }  
}
```




Thanks for your time 😊