**Введение в язык программирования “Java”***(4 пары)*

**Занятие 1-2**

Вступление

1. Компьютер. Из чего он состоит и как работает.

\*ссылка на картинку <http://informat444.narod.ru/COMP/index.htm> <http://informat444.narod.ru/COMP/ris/shema.JPG>

\*ссылка на презентацию <http://www.myshared.ru/slide/1226814/>

2. Операционная система. Для чего она нужна.

\* ссылка на картинку <http://dic.academic.ru/pictures/wiki/files/52/400px-GeneralizedLayeredComputerStructure_OS.png>

\* ссылка на презентацию <http://www.myshared.ru/slide/114210/>

3. Командная строка. Интерпретатор командной строки. Интерфейс командной строки. Формат команд. Привести примеры команд (всем вместе поделать).

\*\*\*Здесь можно порекомендовать источники, такие как Брюс Эккель «Философия Java», И. Н. Блинов «Промышленное программирование», лекции на ютубе Мирончик, Ткач и др.\*\*\*

4. История и этапы развития языка “Java”. Историю, в том числе и про кофе, все и сами могут почитать. Здесь можно рассмотреть историю версий и набор технологий java.

\* ссылка на картинку <https://hsto.org/getpro/habr/post_images/7fa/f43/cdb/7faf43cdbc812f0accc5a4e90c9eb0f5.png>

\* ссылка на картинку <http://images.myshared.ru/6/653665/slide_9.jpg>

5. Сравнительный анализ языка “Java” с другими языками программирования

\* ссылка на картинку <https://hsto.org/getpro/habr/post_images/55e/bf4/e0d/55ebf4e0dd9c9ae6ae0cca6d16dcdac0.png>

6. Что такое виртуальная машина? Здесь можно рассмотреть такие понятия, как интерепретация, трансляция, компиляция. Что такое виртуальная машина вообще и JVM в частности.

\* ссылки (на выбор)

<http://img.viralpatel.net/2008/12/java-program-execution.png>

<https://image.slidesharecdn.com/jvm-140726005513-phpapp01/95/jvm-7-638.jpg?cb=1406336216>

<http://images.myshared.ru/5/414179/slide_4.jpg>

И вот здесь можно полазить много где, посмотреть, попрактиковаться с командной строкой, например.

7. Что такое байт-код? В целом и байт-код java в частности. Что это такое, зачем нужен, какие преимущества, в чём отличие от машинного кода.

Алгоритм

\*\*\*Алгоритмы и блок схемы подробно прошли на предварительных занятиях, решили много заданий. Наверно, стоит кратко освятить эти две темы\*\*\*

1. Понятие алгоритма. Определение. Можно в историю немного заглянуть, про машину Тьюринга рассказать, например. Свойства алгоритмов:

* [Дискретность](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C) — алгоритм должен представлять процесс решения задачи как последовательное выполнение некоторых простых шагов. При этом для выполнения каждого шага алгоритма требуется конечный отрезок времени, то есть преобразование исходных данных в результат осуществляется во времени дискретно.
* [Детерминированность](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C) (определённость). В каждый момент времени следующий шаг работы однозначно определяется состоянием системы. Таким образом, алгоритм выдаёт один и тот же результат (ответ) для одних и тех же исходных данных. В современной трактовке у разных реализаций одного и того же алгоритма должен быть изоморфный [граф](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B0). С другой стороны, существуют вероятностные алгоритмы, в которых следующий шаг работы зависит от текущего состояния системы и генерируемого случайного числа. Однако при включении метода генерации случайных чисел в список «исходных данных» вероятностный алгоритм становится подвидом обычного.
* Понятность — алгоритм должен включать только те команды, которые доступны исполнителю и входят в его систему команд.
* Завершаемость (конечность) — в более узком понимании алгоритма как математической функции, при правильно заданных начальных данных алгоритм должен завершать работу и выдавать результат за определённое число шагов.[[](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%A1%D1%81%D1%8B%D0%BB%D0%BA%D0%B8_%D0%BD%D0%B0_%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8" \o "Википедия:Ссылки на источники)*[источник не указан 2310 дней](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%A1%D1%81%D1%8B%D0%BB%D0%BA%D0%B8_%D0%BD%D0%B0_%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8" \o "Википедия:Ссылки на источники)*[]](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%A1%D1%81%D1%8B%D0%BB%D0%BA%D0%B8_%D0%BD%D0%B0_%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8" \o "Википедия:Ссылки на источники) С другой стороны, вероятностный алгоритм может и никогда не выдать результат, но вероятность этого равна 0.
* Массовость (универсальность). Алгоритм должен быть применим к разным наборам начальных данных.
* Результативность — завершение алгоритма определёнными результатами.
* Алгоритм содержит ошибки, если приводит к получению неправильных результатов либо не даёт результатов вовсе.
* Алгоритм не содержит ошибок, если он даёт правильные результаты для любых допустимых начальных данных.

2. Примеры использования алгоритмов в реальной жизни. Засунуть слона в холодильник, сходить в магазин за пивом, отпраздновать день рождения, посчитать сумму нескольких чисел и многое другое.

3. Типы алгоритмов.

Линейный - набор команд (указаний), выполняемых последовательно во времени друг за другом (посадить дерево, заварить чай).

Разветвлённый - алгоритм, содержащий хотя бы одно условие, в результате проверки которого может осуществляться разделение на несколько альтернативных ветвей алгоритма (сбор грибов).

Циклический - алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов. Цикл программы — последовательность команд (серия, тело цикла), которая может выполняться многократно (для новых исходных данных) до удовлетворения некоторого условия (сбор грибов-опят, мытьё посуды).

Понятие блок-схемы

1. Базовые обозначения в блок-схемах

2. Блок начала алгоритма

3. Блок завершения алгоритма

4. Блок ввода данных

5. Блок вывода данных

6. Блок вычислений

7. Простейшие примеры использования блок-схем

Программная среда “Eclipse”

\*\*\*Установить, всё, что надо рассказать, показать, посоздавать, поудалять с активным участием студентов\*\*\*

1. Инсталляция

2. Основы работы с IDE Eclipse

3. Создание проекта

4. Добавление файла к проекту

5. Обзор альтернативных средств разработки

* [JDK](https://ru.wikipedia.org/wiki/JDK) — помимо набора библиотек для платформ [Java SE](https://ru.wikipedia.org/wiki/Java_SE" \o "Java SE) и [Java EE](https://ru.wikipedia.org/wiki/Java_EE" \o "Java EE), содержит компилятор командной строки javac и набор утилит, также работающих в режиме командной строки.
* [NetBeans IDE](https://ru.wikipedia.org/wiki/NetBeans_IDE) — свободная интегрированная среда разработки для всех платформ Java — [Java ME](https://ru.wikipedia.org/wiki/Java_ME" \o "Java ME), [Java SE](https://ru.wikipedia.org/wiki/Java_SE" \o "Java SE) и [Java EE](https://ru.wikipedia.org/wiki/Java_EE" \o "Java EE). Пропагандируется [Oracle](https://ru.wikipedia.org/wiki/Oracle" \o "Oracle), владельцем технологии Java, как базовое средство для разработки ПО на языке Java и других языках ([C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), [C++](https://ru.wikipedia.org/wiki/C%2B%2B), [PHP](https://ru.wikipedia.org/wiki/PHP), [Fortran](https://ru.wikipedia.org/wiki/Fortran" \o "Fortran) и др.).
* [Eclipse IDE](https://ru.wikipedia.org/wiki/Eclipse_(%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8)) — свободная интегрированная среда разработки для [Java SE](https://ru.wikipedia.org/wiki/Java_SE" \o "Java SE), [Java EE](https://ru.wikipedia.org/wiki/Java_EE" \o "Java EE) и [Java ME](https://ru.wikipedia.org/wiki/Java_ME" \o "Java ME)[[36]](https://ru.wikipedia.org/wiki/Java#cite_note-37). Пропагандируется [IBM](https://ru.wikipedia.org/wiki/IBM), одним из важнейших разработчиков корпоративного ПО, как базовое средство для разработки ПО на языке Java и других языках ([C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), [C++](https://ru.wikipedia.org/wiki/C%2B%2B), [Ruby](https://ru.wikipedia.org/wiki/Ruby" \o "Ruby), [Fortran](https://ru.wikipedia.org/wiki/Fortran" \o "Fortran) и др.).
* [IntelliJ IDEA](https://ru.wikipedia.org/wiki/IntelliJ_IDEA) — среда разработки для платформ [Java SE](https://ru.wikipedia.org/wiki/Java_SE" \o "Java SE), [Java EE](https://ru.wikipedia.org/wiki/Java_EE" \o "Java EE) и [Java ME](https://ru.wikipedia.org/wiki/Java_ME" \o "Java ME). Разработчик — компания [JetBrains](https://ru.wikipedia.org/wiki/JetBrains" \o "JetBrains). Распространяется в двух версиях: свободной бесплатной (Community Edition) и коммерческой проприетарной (Ultimate Edition).
* [JDeveloper](https://ru.wikipedia.org/wiki/JDeveloper) — среда разработки для платформ [Java SE](https://ru.wikipedia.org/wiki/Java_SE" \o "Java SE), [Java EE](https://ru.wikipedia.org/wiki/Java_EE" \o "Java EE) и [Java ME](https://ru.wikipedia.org/wiki/Java_ME" \o "Java ME). Разработчик — компания [Oracle](https://ru.wikipedia.org/wiki/Oracle" \o "Oracle).
* [BlueJ](https://ru.wikipedia.org/wiki/BlueJ) — среда разработки программного обеспечения на языке Java, созданная в основном для использования в обучении, но также подходящая для разработки небольших программ.
* [Java для iOS](https://ru.wikipedia.org/w/index.php?title=Java_%D0%B4%D0%BB%D1%8F_iOS&action=edit&redlink=1) — обучающее приложение и компилятор для [iOS](https://ru.wikipedia.org/wiki/IOS" \o "IOS).
* [Geany](https://ru.wikipedia.org/wiki/Geany) — [свободная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [среда разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), написанная с использованием [библиотеки](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%B8_%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F) [GTK2](https://ru.wikipedia.org/wiki/GTK2).

6. Запуск простейшего приложения. Конечно же «Hello, World!» И надо сказать, почему именно его и что это значит для любого программиста.

\*Если переходить к типам переменных, то надо дать определение переменной и рассмотреть типы в java. Что такое объявление переменной и что это значит. И для практики попреобразовывыть одни в другие.

|  |  |  |
| --- | --- | --- |
| **Тип** | **Длина (в байтах)** | **Диапазон или набор значений** |
| boolean | 1 в массивах, 4 в переменных | true, false |
| byte | 1 | −128..127 |
| char | 2 | 0..216−1, или 0..65535 |
| short | 2 | −215..215−1, или −32768..32767 |
| int | 4 | −231..231−1, или −2147483648..2147483647 |
| long | 8 | −263..263−1, или примерно −9.2·1018..9.2·1018 |
| float | 4 | -(2-2−23)·2127..(2-2−23)·2127, или примерно −3.4·1038..3.4·1038, а также {\displaystyle -\infty }, {\displaystyle \infty }, NaN |
| double | 8 | -(2-2−52)·21023..(2-2−52)·21023, или примерно −1.8·10308..1.8·10308, а также {\displaystyle -\infty }, {\displaystyle \infty }, NaN |

Список простых задач (линейные алгоритмы)

1. Два светила находятся на данном расстоянии d друг от друга, движутся одно к одному с данными скоростями V1,V2.Определить точку их встречи.
2. Определить время, через которое встретятся два тела, равноускоренно движущиеся навстречу друг другу, если известны их начальные скорости v1 и v2, ускорения а1 и а2 и начальное расстояние между ними.
3. Найти площадь квадрата, если две его противоположные вершины заданы координатами (x1,y1), (x2,y2).
4. Поменять значение двух переменных: а) используя дополнительную переменную; б) не используя дополнительной переменной.
5. В трех коробках А1, А2, А3 находятся соответственно 11, 7 и 6 спичек. Составьте алгоритм перемещения спичек из одной коробки в другую с целью выравнивания количества спичек во всех коробках. Добавлять в коробку можно лишь столько спичек, сколько в ней находится.
6. Два лица имеют равные капиталы, причём каждый состоит из известного числа вещей (a,b) одинаковой ценности и известного числа монет (n,m). Но как число вещей, так и сумма денег у каждого различны. Какова ценность вещи.
7. Определить длину сторон прямоугольника, если известно отношение сторон как m:n и площадь фигуры S.
8. Определить высоту треугольника, если даны три его стороны (a,b,c).
9. Вычислить в равностороннем треугольнике сторону, высоту и площадь, если радиус вписанной окружности равен r.
10. Найти площадь прямоугольника, основание которого в n раз больше высоты, а площадь численно равна периметру.
11. Разделить число m на такие две части, разность которых есть 5.
12. Отец завещал 1/3 своего имения сыну и 2/5 дочери; из оставшегося затем капитала n руб. должны пойти на уплату долга, а m руб. в пользу вдовы. Как велик был оставленный отцом капитал и поскольку должен получить сын и дочь.
13. Некто на вопрос о возрасте двух его сыновей отвечал: "Первый мой сын втрое старше второго, а обоим им вместе столько лет, сколько мне было m лет тому назад; мне теперь n лет". Найти возраст обоих сыновей.
14. Купец купил q аршин чёрного и синего сукна за b руб. Спрашивается, сколько аршин бы купил того и другого, если синее сукно стоило n руб. за аршин, а чёрное - m руб.
15. Дано натуральное число N. Составить алгоритм для получения последней цифры этого числа.
16. Дано натуральное трёхзначное число N. Составить алгоритм для получения суммы цифр этого числа.
17. Дано натуральное трёхзначное число N. Составить алгоритм для получения числа M, являющегося перевёртышем числа N; например: N=123,M=321.
18. Даны действительные числа X,Y,Z. Составить алгоритм, определяющий среднее арифметическое и среднее геометрическое их модулей.
19. Известна длина окружности L. Составить алгоритм, определяющий площадь круга, ограниченного этой окружностью.
20. Даны длины катетов прямоугольного треугольника (a,b). Составить алгоритм, определяющий его гипотенузу, площадь, радиус описанной окружности.
21. Даны две стороны треугольника (a,b) и угол между ними C. Определить третью сторону, площадь и радиус описанной окружности.
22. Найти центр окружности, если даны координаты трёх точек на ней.
23. Найти координаты четвёртой вершины прямоугольника, если даны координаты трёх его вершин.
24. Определить гипотенузу прямоугольного треугольника, если известны его площадь и периметр (S,P).