Основы программирования на языке Java. Уровень 1

Степулёнок Денис Олегович — denis.stepulenok@oracle.com
Солодкая Анастасия Сергеевна — a.s.solodkaya@gmail.com

7 октября 2016 года

Опрос слушателей курса I

Я Денис Олегович Степулёнок, работаю Java-программистом в компании Oracle (старший инженер-программист).
Солодкая Анастасия Сергеевна — тоже Java-программист в компании Oracle.
Сейчас я проведу краткий опрос чтобы откорректировать курс с учётом ваших пожеланий.
Результаты запишем в Google Docs о курсе `https://docs.google.com/document/d/1ntLJl5sXYvjBe4VPylm8Q7KHO5ki18Oz1QYOwLJ9-E4/edit`, чтобы по окончанию курса свериться по этой таблице. Её мы посмотрим на последнем занятии чтобы оценить ваш прогресс.

- Имя (как вас называть)
- Как вы оцениваете ваш текущий уровень программирования (Java)?
- Ваши цели и ожидания от курса?
- По окончанию курса мы предложим вам оценить свои достижения (результаты)

Программа (агенда) курса I

- Знакомство. Java SE: JRE, JDK, JVM. Установка IntelliJ IDEA. Создание первого проекта (метод main)
- Переменные и примитивные типы данных, арифметические операции
- Управляющие конструкции: if, else. Преобразование типов. Логические операции
- Массивы. Циклы. Базовые алгоритмы поиска. Сортировка
- Введение в классы и объекты.
- Коллекции (java.util): ArrayList, LinkedList, HashSet, HashMap, TreeSet
- ООП: четыре кита, геттеры и сеттеры, this, конструкторы
- Абстрактные классы и методы. Интерфейсы. Анонимные классы
- Параметризация. Лямбда-выражения
- Рекурсия (рекурсивные алгоритмы)
- Работа с файлами. Обработка исключений
- Совместная работа над проектом

Перед тем, как мы начнём: I

- Добавляйтесь в «Телегу», наш чат в Telegram: `https://telegram.me/joinchat/AGxRsz72CHo9cH-u89Ca1g`
- Зарегистрируйтесь на `https://github.com` — у вас появится логин на github.
- Пришлите свой github-логин мне на denis.stepulenok@oracle.com и я добавлю вас в группу `https://github.com/levelp`
- Скачайте последнюю версию JavaSE: `http://www.oracle.com/technetwork/java/javase/downloads/index.html`
- Скачайте Idea Ultimate Edition: `https://www.jetbrains.com/idea/download`
- Зарегистрируйтесь на сайте JetBrains: `https://account.jetbrains.com/login` — для регистрации Idea Ultimate

План (агенда) сегодняшнего занятия I

- Что такое программа? Чем занимается программист?
- Языков программирования много (слайд со списком)
- Какое место занимает Java? Почему стоит программировать на Java?
- Парадигмы программирования: структурное, ООП, функциональное...
- Процесс создания приложения
- Создать класс в блокноте, скопилировать и запустить их из командной строки.
- Знакомство с Java SE - JDK, JRE.. ME/SE/EE
- Виртуальная машина Java - JVM
- Среды разработки: Idea, Netbeans, Eclipse...
- Среда разработки IntelliJ IDEA
- Создание первого проекта
- Метод main, psvm - сокращения клавиатурные
- Команды вывода в консоль - print, println, печать разных типов

План (агенда) сегодняшнего занятия II

- Экранирование символов
- Переменные и примитивные типы данных 8 примитивных + показываем как они объявляются
- Арифметические операции - все операции + их приоритеты

Чем занимается программист? I

Чем занимается программист? II

- Из-за того, что я умею пользоваться интернетом, меня многие считают программистом и просят зайти починить их компьютер
- Я — программист. И на вопрос «Кем работаешь?» я отвечал как есть. В результате все мои знакомые и друзья звонят и тащат мне свои системники, ноуты и даже телефоны на починку, просят выбрать «какой-нибудь путёвый» девайс для обновки. Все разъяснения про разницу программиста баз данных и компьютерного слесаря проходят впустую. В один «прекрасный» день мне всё это осточертело. И на вопрос «Кем работаешь?» я стал отвечать «Архитектором баз данных и программных оболочек». Звонки поутихли. Но вчера позвонил приятель и попросил помочь ему спроектировать на даче сортир
- Программист — живой организм, перерабатывающий кофе и пиццу в код
- Программист — конвертер галлюцинаций и фантазий заказчика в жесткую формальную систему понимаемую компьютером
- Общается с заказчиками — `https://www.youtube.com/watch?v=UoKlKx-3FcA` (Видео «Эксперт»)

Чем занимается программист? III

- Потом реализует — https://www.youtube.com/watch?v=B7MIJP90biM (Решение задачи о 7 линиях)
- Чем занимаются программисты — https://www.youtube.com/watch?v=kvIFJNafsC4
- Как оценивать работу программиста? — https://www.youtube.com/watch?v=IHg4Gwgjg_c

Общение с заказчиками I

Общение с заказчиками II

Чем отличается мышление программиста? I

Чем отличается мышление программиста? II

Что такое программа? Чем занимается программист? I

Программа — термин, в переводе означающий «предписание», то есть предварительное описание предстоящих событий или действий. Данное понятие непосредственно связано с понятием «алгоритм».
Программу можно выполнять или дополнять. В программе написаны действия программы и её значение.
Программист — специалист, занимающийся написанием программ для компьютеров.

Чем занимается программист? (спрашиваем и на пальцах рассказываем про процесс) I

«Соберись и разбирайся!»
Что нужно чтобы попасть в Google?

- Java
- Структуры и алгоритмы

Какие проекты написаны на Java? I

- Minecraft `https://minecraft.net`
- Android приложения — возьмите свой телефон на Android, абсолютно все приложения написаны на Java, с использованием Google и Android API
- Вэб-приложения — JSP, RESTful-сервисы, Spring MVC, Struts 2.0
- IDE для программистов (IntelliJ Idea, Eclipse, Netbeans...)
- Серверные приложения в сфере финансовых услуг
- Трейдинговые приложения — большая часть индустрии финансовых услуг
- J2ME приложения
- Встраиваемые системы (холодильники, микроволновки, автомобили)
- Большие данные (Big Data) — Hadoop и другие технологии обработки больших данных так или иначе используют Java, например Hbase и Accumulo от Apache, или ElasticSearch

Какие проекты написаны на Java? II

- Высокочастотные трейдинговые пространства — Java улучшила свои эксплуатационные показатели и с современными JIT-ами она способна предоставить производительность на уровне C++. По этой причине Java популярна и при написании высокопроизводительных систем, потому что хоть производительность проигрывает в сравнении с родным языком, но вы можете пожертвовать безопасностью, мобильностью и надёжностью ради большей скорости и требуется всего один неопытный C++ программист, чтобы сделать приложение медленным и ненадёжным.
- Научные приложения. В наши дни часто Java — выбор по-умолчанию для научных приложений, включая обработку естественного языка. Основная причина в том, что Java более безопасна, мобильна и надёжна и имеет лучшие инструменты параллелизации, чем C++ и другие языки.

Официальная статистика I

Подробнее о технологии Java: `https://www.java.com/ru/about`

- Java используется на 97% корпоративных настольных ПК
- Java используется на 89% настольных ПК в США
- 9 млн разработчиков на Java в мире
- Инструмент номер 1 среди разработчиков
- Программа номер 1 среди разработчиков
- Java используется в 3 млрд мобильных телефонов
- Java входит в комплект поставки 100% всех проигрывателей дисков Blu-ray
- Используется 5 млн Java Card
- Java используется в 125 млн ТВ-устройств
- 5 из 5 основных производителей оригинального оборудования включают в комплект поставки Java ME

Программирование приносит деньги I

Смотрим вакансии на hh.ru / job / superjob

Принципы прагматического программирования I

Позаботьтесь о вашем ремесле Нет смысла разрабатывать программы, если вы не заботитесь о качестве работы.
Думай! О своей работе

Непрерывность процесса I

Во время экскурсии по Итонскому колледжу в Англии турист спросил садовника, как ему удается содержать лужайки в столь идеальном состоянии.
«Это несложно, — ответил садовник, — вы просто стряхиваете росу каждое утро, выкашиваете лужайку через день и утрамбовываете раз в неделю».
«И это все?» — спросил турист.
«Абсолютно все, — ответил садовник, — если заниматься этим на протяжении 500 лет, то ваша лужайка будет не хуже»

Ортогональность I

Ортогональность очень важна, если вы хотите создавать системы, которые легко поддаются проектированию, сборке, тестированию и расширению. Однако этому принципу редко обучают непосредственно. Часто он является лишь скрытым достоинством других разнообразных методик, которые вы изучаете. Это неправильно. Как только вы научитесь непосредственно применять принципы ортогональности, вы сразу заметите, как улучшилось качество создаваемых вами систем.

Что такое ортогональность?

Термин «ортогональность» заимствован из геометрии. Две прямые являются ортогональными, если они пересекаются под прямым углом, например оси координат на графике.

В терминах векторной алгебры две такие линии являются независимыми. Если двигаться вдоль одной из линий, то проекция движущейся точки на другую линию не меняется. Этот термин был введен в информатике для обозначения некой разновидности независимости или несвязанности.

Документация в тексте программы I

TODO: 24 страница с Программист-прагматик Документация в тексте программы ...

- Java Virtual Machine — виртуальная машина Java
- Java — это тоже компилируемый язык программирования. И как у других подобных языков, у него тоже есть стадии написания текста, компиляции и запуска. Но в отличии от языков типа C, Pascal и прочих, программа на Java не сразу компилируется в машинный код. После компиляции создается так называемый байт-код.

История Java I

Немногие языки могут похвастаться тем, что им удалось изменить общее представление о программировании. Но и в этой "элитной"группе один язык выделяется среди прочих. Его влияние очень быстро почувствовали все программисты. Речь, конечно же, идет о Java. Не будет преувеличением сказать, что выпуск в 1995 году компанией Sun Microsystems Inc. версии Java 1.0 вызвал настоящую революцию в программировании. В результате «Всемирная паутина» стала по-настоящему интерактивной средой. Между тем Java установил новый стандарт в разработке языков программирования. Со временем Java усовершенствовался. В отличие от многих других языков, в которых новые средства внедрялись относительно меменно, Java всегда находился на переднем крае разработки языков программирования. Одной из причин, позволивших добиться этого, послужило создание вокруг Java плодотворной атмосферы, способствовавшей внедрению новых идей. В результате язык Java постоянно совершенствовался: одни его изменения были незначительными, а другие - весьма существенными.

Java 1.1 I

Первым существенным обновлением Java стала версия 1.1. Изменения в ней бьmи более значительны, чем это обычно подразумевает переход к новой версии языка программирования. В версии Java 1.1 были добавлены многие библиотечные элементы,
переопределены средства обработки событий, перекомпонованы многие функциональные средства исходной библиотеки версии 1.0.

Java 2 I

Следующим этапом развития данного языка стала платформа Java 2, где цифра 2 означает «второе поколение». Ее появление стало поворотным событием, ознаменовавшим начало «новой эпохи» Java. Первым выпуском Java 2 стала версия 1.2. На первый взгляд, несоответствие номеров в обозначениях Java 2 и версии 1.2 может показаться странным. Дело в том, что номером 1.2 сначала обозначались библиотеки Java и только затем - весь выпуск. Компания Sun перекомпоновала программный продукт Java в J2SE (Java 2 Platform Standard Edition - Стандартная версия платформы Java 2), и с тех пор номера версий стали относиться именно к этому продукту.

J2SE 1.3 I

Затем появилась версия J2SE 1.3, в которую были внесены первые значительные изменения по сравнению с первоначальным выпуском Java 2. Новые функциональные средства были в основном добавлены к уже существующим и более тесно связаны со средой разработки.

J2SE 1.4 I

Версия J2SE 1.4 стала очередным этапом в развитии Java. Она содержала новые важные средства, в том числе цепочки исключений, канальный ввод-вывод и ключевое слово `assert`.

J2SE 5 I

Следующая версия J2SE 5, по сути, стала вторым революционным преобразованием Java. В отличие от большинства предьщущих модернизаций, которые сводились к важным, но предсказуемым усовершенствованиям, в J2SE 5 были существенно расширены рамки применения и функциональные возможности языка, а также повышена его производительность. Для более ясного представления о масштабах изменений, внесенных в версии J2SE 5, ниже дан перечень появившихся возможностей:

- обобщения
- автоупаковка и автораспаковка (boxing / unboxing)
- перечисления
- усовершенствованный вариант цикла for (foreach)
- список аргументов переменной длины
- статический импорт
- аннотации

J2SE 5 II

В этот список не вошли второстепенные дополнения и поэтапные изменения, характерные для перехода к новой версии. Каждый элемент этого списка представляет собой значительное усовершенствование Java. Для поддержки одних нововведений, в том числе обобщений, варианта `for-each` цикла `for` и строки аргументов переменной длины, передаваемой методу, понадобилось ввести новые синтаксические конструкции в язык. Другие нововведения, такие как автоупаковка и автораспаковка, повлияли на семантику языка. И наконец, аннотации открьши совершенно новые возможности для программирования.

Особая значимость описанных новшеств проявилась в том, что новая версия получила номер «5». Можно было ожидать, что номером очередной версии Java будет 1.5. Но нововведения были настолько существенными, что переход от версии 1.4 к 1.5 не отражал бы масштабы внесенных изменений. Поэтому разработчики из компании Sun решили увеличить номер версии до 5, подчеркнув тем самым важность нововведений. В итоге новая версия получила название J2SE 5, а комплект разработчика приложений на языке Java стал называться JDK 5. Но ради согласованности с предыдущими

J2SE 5 III

версиями бьшо решено использовать 1.5 в качестве *внутреннего номера* версии, на который также иногда ссылаются как на номер версии разработки. Цифра «5» в J2SE 5 означает *номер версии программного продукта*.

Java SE 6 I

Следующая версия Java была названа Java SE 6. Это означает, что в компании Sun вновь решили изменить название платформы Java. Прежде всего, из названия исчезла цифра "2". Теперь платформа стала называться Java SE, официальным именем продукта стало Java Platform, Standard Edition 6, а комплект разработчика приложений получил название JDK 6. Как и цифра "5"в названии J2SE 5, цифра "6"в Java SE 6 означает номер версии программного продукта, тогда как внутренним номером версии является 1.6. Версия Java SE 6 создавалась на основе платформы J2SE 5, но отличалась от последней рядом нововведений. Изменения в этой версии не такие масштабные, как в предыдущей, но в ней были улучшены библиотеки интерфейса прикладного программирования (API), добавлен ряд новых пакетов и доработана исполняющая система. По существу, в версии Java SE 6 были закреплены усовершенствования, внедренные в J2SE 5.

Java SE 7 I

Следующая версия Java получила название Java SE 7, а соответствующий комплект разработки приложений - JDK 7. Данной версии присвоен внутренний номер 1.7. Java SE - это первая основная версия Java, выпущенная после того, как компания Sun Microsystems Inc. была приобретена компанией Oracle (этот процесс начался в апреле 2009 года и завершился в январе 2010 года). В версии Java SE 7 появилось немало новых средств, в том числе существенные дополнения были включены как в сам язык, так и в стандартные библиотеки API. Также была усовершенствована исполняющая система Java, в которой теперь поддерживаются программы, написанные на других языках программирования.
Наиболее важные средства, внедренные в версии Java SE 7 и рассматриваемые в этой книге, были разработаны в рамках проекта Project Coin. В этом проекте преследовалась цель определить ряд незначительных изменений в языке Java, которые должны быть внедрены в JDK 7. И хотя эти изменения в целом считаются незначительными, их влияние на процесс разработки программ нельзя недооценивать. На самом деле для многих

Java SE 7 II

программистов они могут стать самыми важными среди всех новых средств, вошедших в Java SE 7. Вот краткий перечень новых языковых средств Java SE 7:

- возможность управления выбором в переключателе swi tch с помощью объектов класса String;
- двоичные целочисленные литералы;
- символы подчеркивания в числовых литералах
- расширенный оператор try, называемый оператором try с ресурсами и поддерживающий автоматическое управление ресурсами (например, теперь файловый поток может быть закрыт, если в нем больше нет необходимости);
- выводимость типов при создании обобщенного экземпляра объекта.

Java SE 7 III

- усовершенствованная обработка исключений, благодаря которой несколько исключений могут быть перехвачены одним (групповым) оператором catch, а также улучшенный контроль типов для повторно генерируемых исключений

Как видите, средства, отнесенные в рамках проекта Project Coin к разряду незначительных языковых изменений, обеспечили преимущества, которые вовсе нельзя считать незначительными. В частности, оператор try с ресурсами существенно сокращает объем кода.

Java SE 8 I

Для новейшей версии Java - Java SE 8 - требуется JDK 8 (Java Development Kit), имеющий внутренний номер версии 1.8. Комплект JDK 8 существенно расширяет возможности языка Java за счет добавления нового языкового средства - лямбда-выражений. Включение в язык лямбда-выражений, изменяющих как концептуальную основу программных решений, так и способ написания кода на Java, будет иметь далеко идущие последствия.
Использование лямбда-выражений позволяет упростить исходный код при создании некоторых языковых конструкций и уменьшить его объем.
Добавление в Java лямбда-выражений привело к появлению в языке нового оператора(->) и нового синтаксического элемента. Эти нововведения лишний раз подтверждают статус Java как живого, развивающегося языка, чего и ожидают от него пользователи.
Помимо лямбда-выражений, в JDK 8 добавлено много новых полезных средств. Так, начиная с JDK 8 стало возможным определять реализации по умолчанию для методов, специфицируемых интерфейсами. Кроме того, в JDK 8 включена поддержка JavaFX - новой библиотеки графического

Java SE 8 II

пользовательского интерфейса (GUI). Ожидается, что вскоре JavaFX станет непременной частью почти всех Java-приложений и практически вытеснит технологию Swing в большинстве GUI-проектов. В завершение можно сказать, что платформа Java SE 8 представляет собой ключевой выпуск, который существенно расширяет возможности языка и вынуждает пересмотреть подходы к написанию кода на Java. Влияние описанных нововведений будет ощущаться всеми разработчиками Java-программ еще протяжении многих лет.
`https://ru.wikipedia.org/wiki/Java`

- Java Virtual Machine — виртуальная машина Java
- Java — это тоже компилируемый язык программирования. И как у других подобных языков, у него тоже есть стадии написания текста, компиляции и запуска. Но в отличии от языков типа C, Pascal и прочих, программа на Java не сразу компилируется в машинный код. После компиляции создается так называемый байт-код.

Процесс создания приложения I

«Водопадная модель»

- Анализ требований (Спецификация программного обеспечения)
- Проектирование программного обеспечения
- Программирование
- Тестирование программного обеспечения
- Системная интеграция (System integration)
- Внедрение программного обеспечения (или Установка программного обеспечения)
- Сопровождение программного обеспечения

Экстремальное программирование (XP) I

- Короткий цикл обратной связи (Fine-scale feedback)
  - Разработка через тестирование (Test-driven development)
  - Игра в планирование (Planning game)
  - Заказчик всегда рядом (Whole team, Onsite customer)
  - Парное программирование (Pair programming)
- Проектирование программного обеспечения
- Программирование
- Тестирование программного обеспечения
- Системная интеграция (System integration)
- Внедрение программного обеспечения (или Установка программного обеспечения)
- Сопровождение программного обеспечения

Среды разработки: Idea, Netbeans, Eclipse I

Среда разработки IntelliJ IDEA I

План практики I

- Скачиваем, устанавливаем и настраиваем JDK
- Собираем и запускаем первое приложение из командной строки
- Скачиваем и устанавливаем Intellij Idea
- Создаем первый проект в Intellij Idea
- Решаем линейное уравнение

Скачиваем, устанавливаем и настраиваем JDK I

- Скачиваем, устанавливаем и настраиваем JDK
- Собираем и запускаем первое приложение из командной строки
- Скачиваем и устанавливаем Intellij Idea

Первая программа на Java I

```java
package java_01; // Имя пакета
public class Main { // Класс «Main»
    // Основной метод - запуск программы
    // psvm
    public static void main(String[] args) {
        // Выводим строчку на экран
        System.out.println("Hello, world!");
    }
}
```

Компилируем из командной строки: `javac Main.java`

Запускаем: `java Main`

Возможности Intellij Idea I

- Скачиваем, устанавливаем и настраиваем JDK
- Собираем и запускаем первое приложение из командной строки
- Скачиваем и устанавливаем Intellj Idea

Создание проекта в Intellij Idea I

- Скачиваем, устанавливаем и настраиваем JDK
- Собираем и запускаем первое приложение из командной строки
- Скачиваем и устанавливаем Intellij Idea

Переменная I

Переменная — именованная ячейка памяти, в которой можно хранить присваиваемое значение. В процессе выполнения программы значение переменной может изменяться. Это означает, что содержимое переменной не фиксировано.

Метод main, psvm - сокращения клавиатурные I

psvm

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

Команды вывода в консоль - print, println, печать разных типов I

Экранирование символов I

Переменные и примитивные типы данных 8 примитивных + показываем как они объявляются I

В Java 8 примитивных типов, которые делят на 3 группы:

- 1. Целые числа - byte, short, char, int, long
- 2. Числа с плавающей точкой - float, double
- 3. Логический - boolean

Целые числа: byte, short, char, int, long I

- Тип — Размер (бит) — Диапазон
- byte 8 бит $-128..127$
- short 16 бит $-32768..32767$
- char 16 бит $0..65535$ — символ
- int 32 бит $-2147483648..2147483647$
- long 64 бит $-9223372036854775808..9223372036854775807$

Типы с плавающей точкой I

- float 32 бит — от $-1.4e-45f$ до $3.4e+38f$
  $1.40129846 * 10^{-45}..3.40282347 * 10^{+38}$
- double 64 бит — от $-4.9e-324$ до $1.7+308$
  $4.94065646 * 10^{-324}..1.79769313 * 10^{+308}$

```
public class FloatingPointTypes {
  public static void main(String[] args) {
    double a, b = 4.12;
    a = 22.1 + b;
    float pi = 3.14f; // При использовании типа float требуется указывать суффикс f или F
    // так как дробные литералы типа double
    float anotherPi = (float) 3.14; // Можно привести явно
    double c = 27;
    double d = pi * c;
    System.out.println(d);
  }
}
```

Арифметические операции - все операции + их приоритеты I

- $+$ Сложение (а также унарный плюс)
- $-$ Вычитание (а также унарный минус)
- $*$ Умножение
- $/$ Деление
- $\%$ Деление по модулю (остаток от деления)
- $++$ Инкремент
- $--$ Декремент

Методы тестирования и отладки I

- Ручное тестирование / ручная отладка — Steps to reproduce / шаги "как воспроизвести недостаток - слишком много ресурсов на повторение шагов каждый раз
- Использование отладчика (debugger)
- Логгирование (протоколирование) — Единственный, который доступен на компьютере конечного пользователя
- Автоматические тесты
  - Модульные: Unit-тесты
  - Функциональные:
    - По спецификации
    - Регрессионные тесты
  - Интеграционные тесты (integration)
  - Тесты производительности (perfomance)

Перед тем, как мы начнём: I

- Подключитесь к конференции в Skype
- Зарегистрируйтесь на `https://github.com` — у вас появится логин на github.
- Пришлите свой github-логин мне на denis.stepulenok@oracle.com и я добавлю вас в группу `https://github.com/levelp`
- Скачайте последнюю версию JavaSE: `http://www.oracle.com/technetwork/java/javase/downloads/index.html`
- Скачайте Idea Ultimate Edition: `https://www.jetbrains.com/idea/download`
- Зарегистрируйтесь на сайте JetBrains: `https://account.jetbrains.com/login` — для регистрации Idea Ultimate
- Чат в Telegram: `https://telegram.me/joinchat/AGxRsz72CHo9cH-u89Ca1g`

Сайты и ссылки I

- http://vk.com/java_course — группа для участников курса
- http://levelp.ru/courses/programmirovanie/java-junior-developer — курс Java Junior Developer на сайте LevelUp
- http://levelp.ru/courses/programmirovanie/basics-of-programming-in-the-java-language-level-1 — «Java - уровень 1»
- http://levelp.ru/courses/programmirovanie/basics-of-programming-in-the-java-language-level-2 — «Java - уровень 2»
- http://javarush.ru — курс Java для начинающих (http://info.javarush.ru/page/learning_plan — план обучения)
- http://www.intuit.ru/studies/courses/16/16/info - "Программирование на Java"курс лекций на ИНТУИТ
- http://hashcode.ru - ответы на вопросы

История и факты I

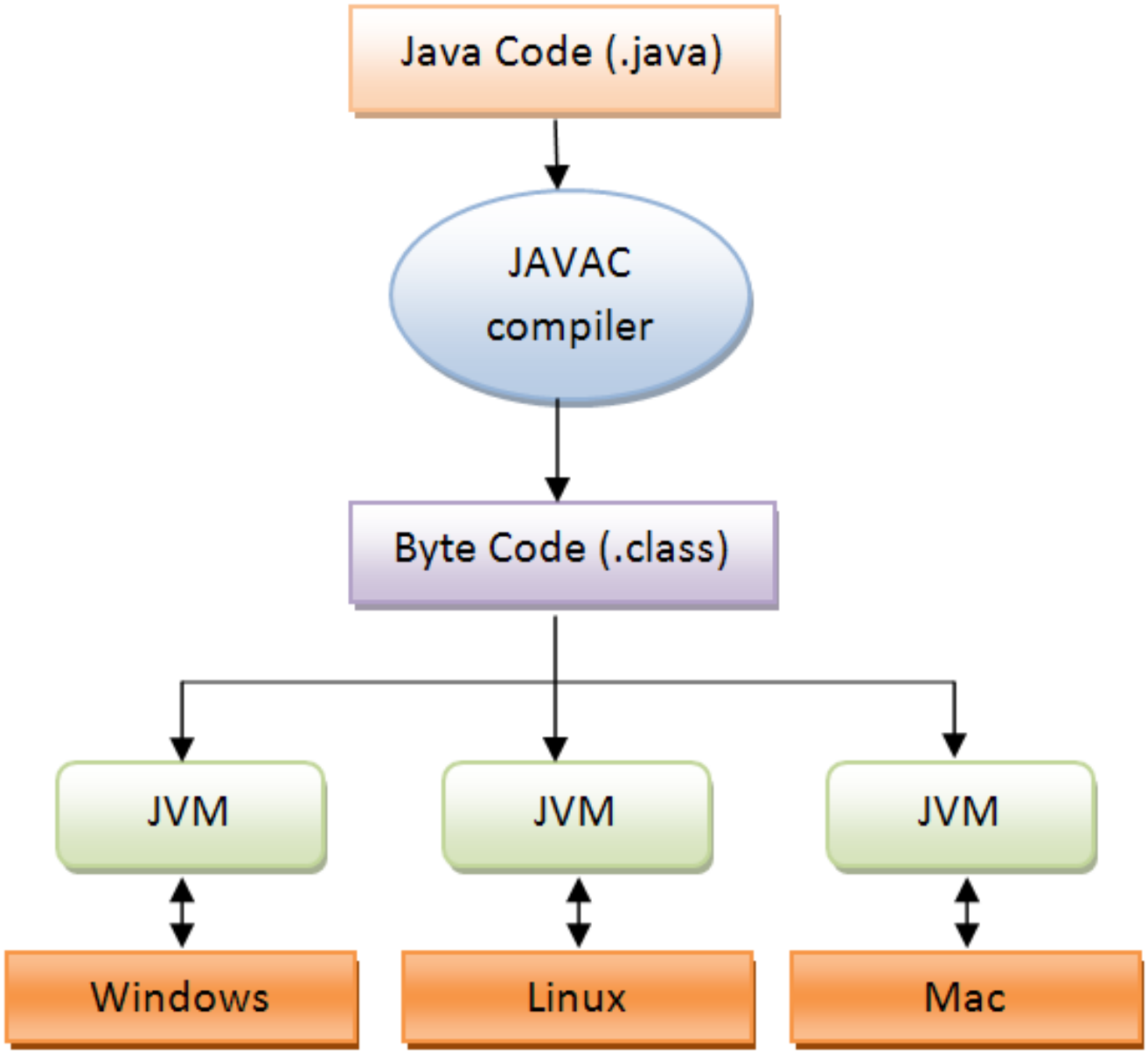
- 1991 — началась работа над языком
- 1995 — первая версия языка выпущена компанией Sun
- WORA — Write once, run everywhere
- 1998 — выпуск Java2: редакции J2ME, J2SE, J2EE
- 1998 — внедрен Java Community Process
- 2007 — Java лицензируется под GNU General Public License
- Jan 2010 - Компания Sun Microsystems поглощается Oracle

Байт-код I

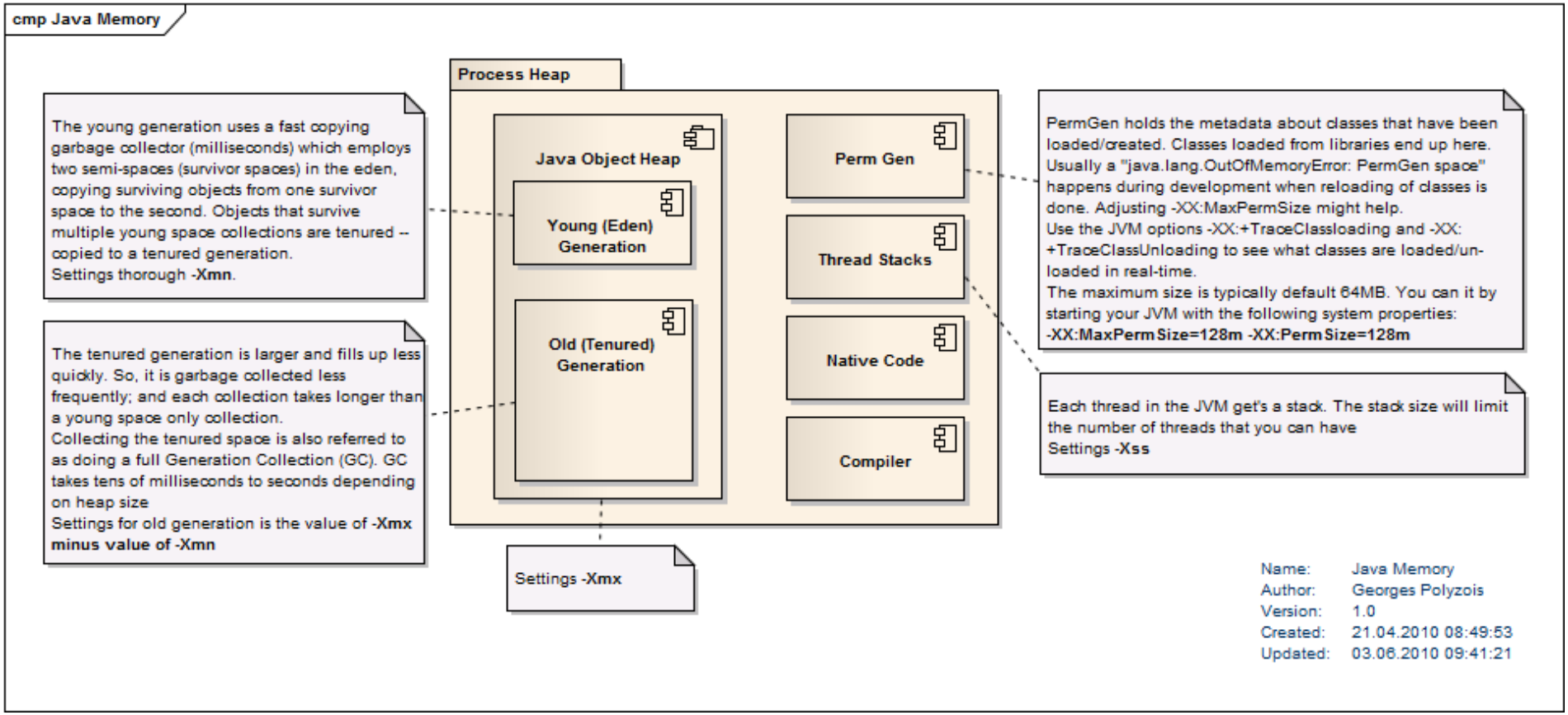
Байт-код (bytecode) — промежуточное представление, в которое компилируется Java-код.
По сравнению с исходным кодом, удобным для создания и чтения человеком, байт-код — компактное представление программы, уже прошедшей синтаксический и семантический анализ. В нём в явном виде закодированы типы, области видимости...
Байт-код — машинно-независимый код низкого уровня (который интерпретируется, либо компилируется в машинные инструкции конкретного процессора).

JVM и ByteCode I

JVM и ByteCode II

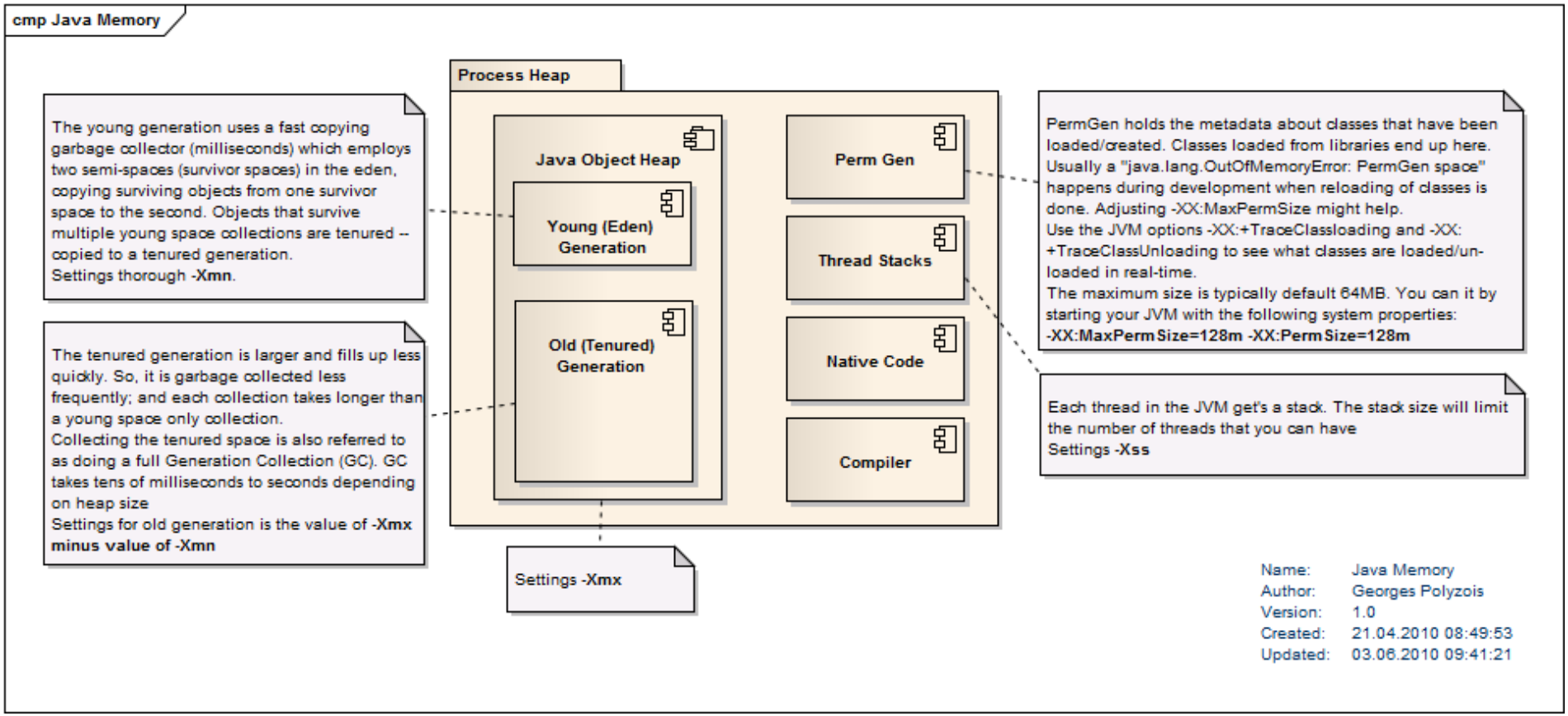


Java Memory Model I



Java Memory Model I

TODO: примитивные типы



Вывод всех правильных скобочных последовательностей I

```
// Вывод всех правильных скобочных последовательностей
public class Brackets {
    int n;
    public Brackets(int n) { this.n = n; }

    void genBrackets(int open, int close, String s) {
        if (open == n && close == n) {
            System.out.println(s);
            return;
        }
        if (open < n) genBrackets(open + 1, close, s + "(");
        if (close < open) genBrackets(open, close + 1, s + ")");
    }

    public static void main(String[] args) {
        for (int n = 1; n < 6; n++) {
            // n - количество открывающих и закрывающих скобок
            System.out.println("n = " + n);
            Brackets generator = new Brackets(n);
            generator.genBrackets(0, 0, "");
        }
    }
```

Вывод всех правильных скобочных последовательностей II

```
}
```

Exceptions: checked / unchecked I

Исключения делятся на несколько классов, но все имеют общего предка — класс **Throwable**.

- Exception — контролируемые исключения (checked)
- Неконтролируемые исключения (unchecked)
  - RuntimeException
  - Errors —

Обработка исключений I

```
try {
  throw new RuntimeException();
} catch (Exception e) {
}
```

Что такое контроль версий, и зачем он вам нужен? I

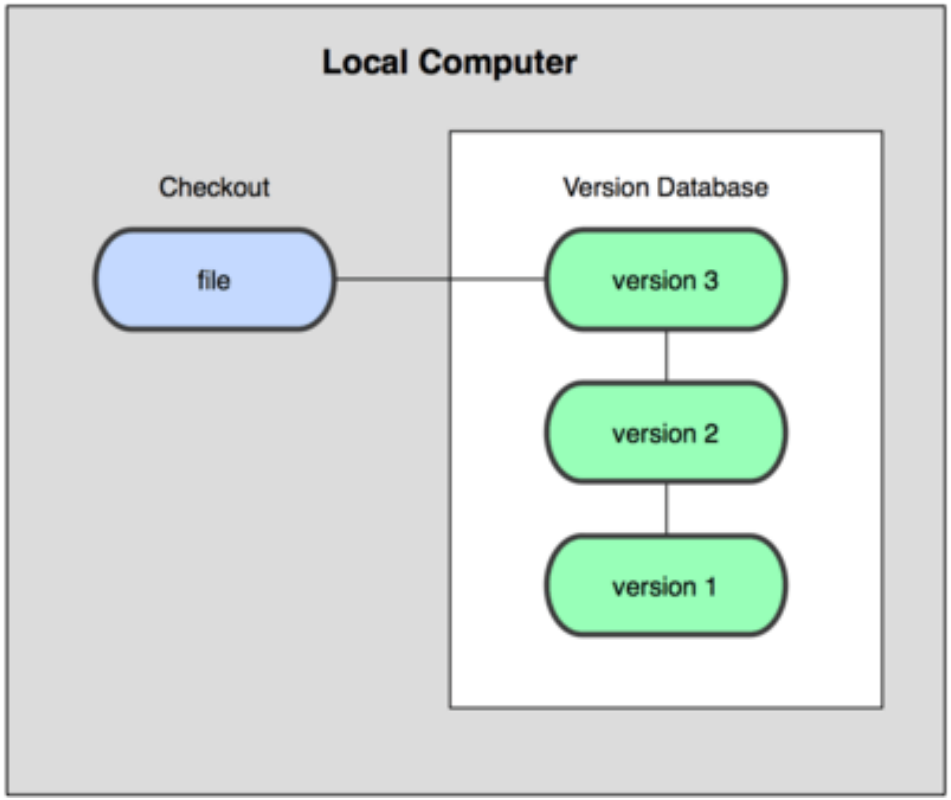
Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов. СКВ даёт возможность возвращать отдельные файлы к прежнему виду, возвращать к прежнему состоянию весь проект, просматривать происходящие со временем изменения, определять, кто последним вносил изменения во внезапно переставший работать модуль, кто и когда внёс в код какую-то ошибку, и многое другое. Вообще, если, пользуясь СКВ, вы всё испортите или потеряете файлы, всё можно будет легко восстановить.
Системы контроля версий:

- Централизованные
- Децентрализованные

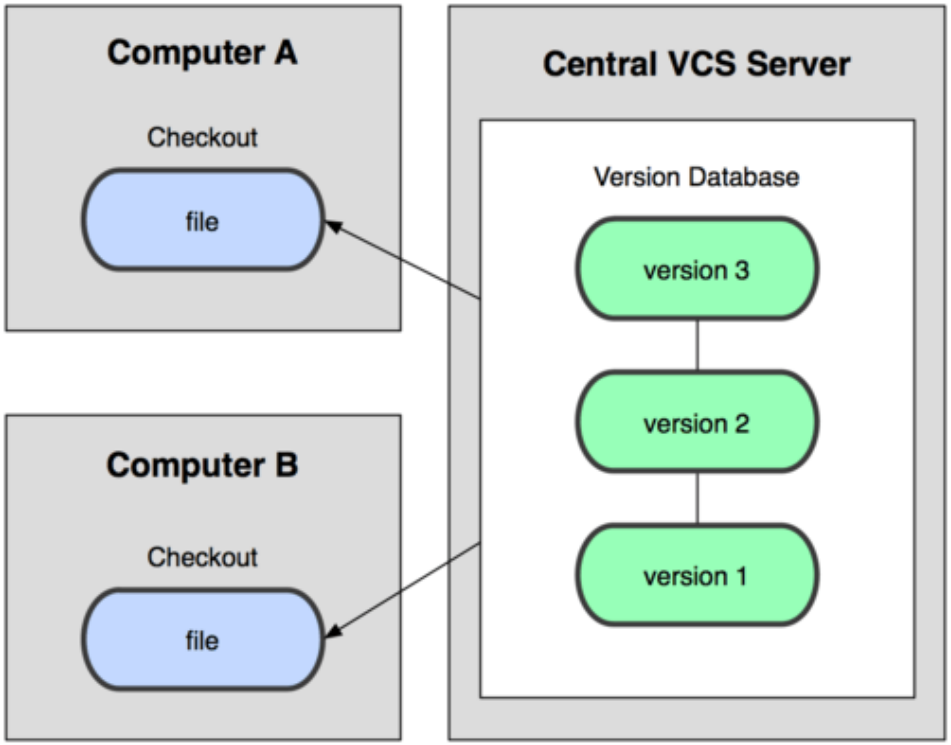
Subversion I

Бесплатная система управления версиями с открытым исходным кодом. Subversion позволяет управлять файлами и каталогами, а так же сделанными в них изменениями во времени. Это позволяет восстановить более ранние версии данных, даèт возможность изучить историю всех изменений. Благодаря этому многие считают систему управления версиями своего рода «машиной времени».

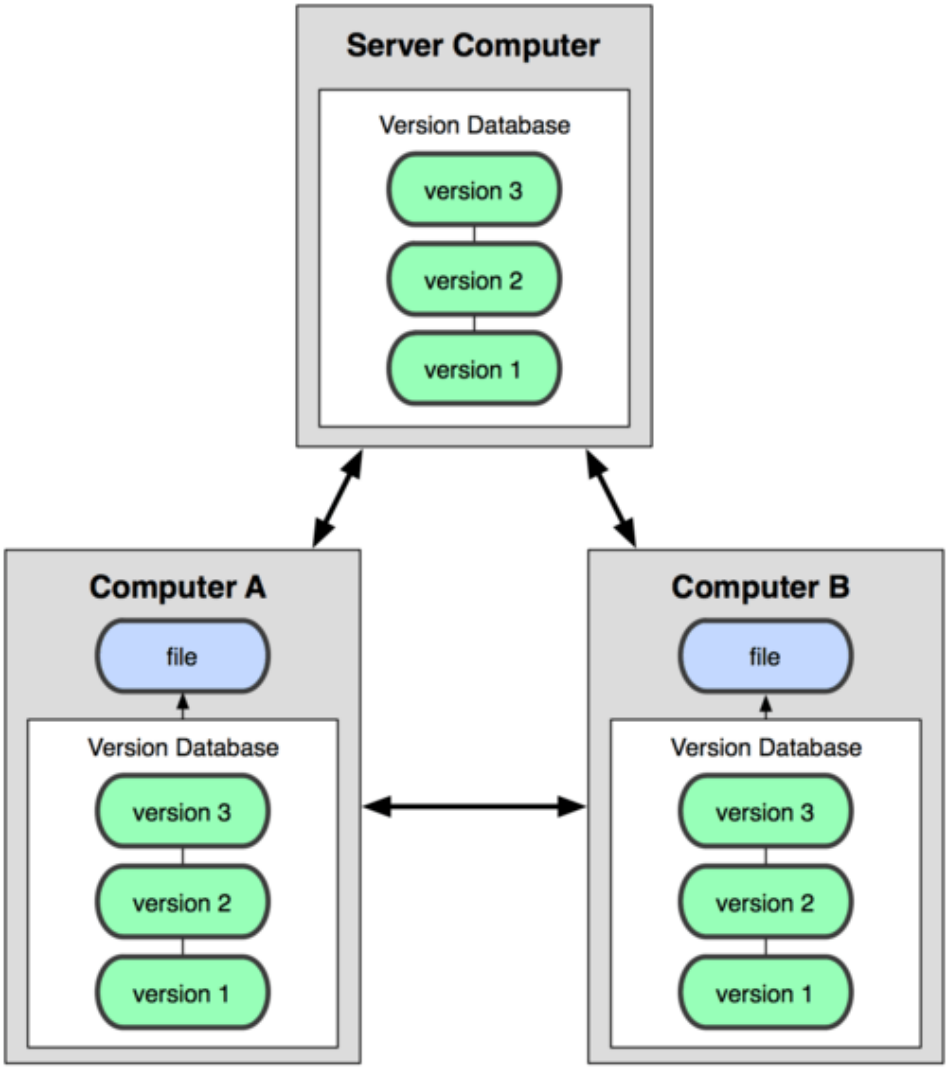
Subversion II



Централизованные системы контроля версий I



Распределённые системы контроля версий I



Git I

Распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано.
Примерами проектов, использующих Git, являются: ядро Linux, Android, Drupal, Cairo, GNU Core Utilities, Mesa, Wine, Chromium, Compiz Fusion, FlightGear, jQuery, PHP, NASM, MediaWiki, DokuWiki, Qt и некоторые дистрибутивы Linux

Откуда скачать и установить Git? I

Для Windows — `https://git-for-windows.github.io`
Нажимаете **Download**, скачивается исполняемый файл, например **Git-2.6.1-64-bit.exe**
Устанавливаете его.
После установки должна появится в командной строке команда: `git`

Основные команды Git I

- `git init` — создание нового git-репозитория в текущей папке. С этого момента в этой папке появляется «машина времени» для изменений. Если вы сделали изменение и зафиксировали его, вы всегда можете откатиться к этой версии. При успешной инициализации выводится сообщение: `Initialized empty Git repository in FolderName`
- `git add` — добавление файлов в систему контроля версий. Например: `git add a.cpp` — добавить файл a.cpp в СКВ.
- `git add *.cpp` — добавить все .cpp файлы в СКВ.
- `git add .` — добавить все файлы и все каталоги с подкаталогами в СКВ.
- `git commit -a -m` — фиксация изменений.
- `git push` — фиксация изменений.
- `git pull` — получить изменения с удалённого сервера.

Контрольные вопросы: I

- Сколько примитивных типов в Java?
- Какие вы знаете целые типы?
- Какие типы с плавающей точкой?
- Арифметические операции?