***1***

1. Just-In-Time-компилятор позволяет

2. Для разработки программ на языке Java необходима установка

3. Запуск Java-программы производится

4. Использование языковой виртуальной машины позволяет добиться

5. Компиляция исходного кода производится

6. Особенностью языка программирования Java является

7. Пакеты служат для

8. Сигнатура функции main имеет вид

9. Синтаксис языка программирования Java схож с синтаксисом языка

10. Структурными частями простейшего Java-проекта являются

11. Точкой входа в любую Java-программу является

12. Язык программирования Java является

***2***

13. В выражении, содержащем переменные типа byte, short, char происходит автоматическое повышение типа до

14. В выражении, содержащем переменные типа double и long, происходит автоматическое повышение типа до

15. В выражении, содержащем переменные типа float и double, происходит автоматическое повышение типа до

16. В выражении, содержащем переменные типа float, происходит автоматическое повышение типа до

17. В выражении, содержащем переменные типа int и long, происходит автоматическое повышение типа до

18. К классам-оберткам над элементарными типами данных не относится

19. К типам с плавающей запятой относится тип

20. К целочисленным типам данных Java не относится тип

***3***

21. В Java ссылки всегда передаются

22. Значение null используется в Java

23. Объект может быть передан по ссылке

24. Объект называется достижимым, если

25. Преобразование типа переменной подкласса к типу суперкласса

26. Преобразование типа переменной суперкласса к типу подкласса

27. Приведение ссылочных типов возможно

28. Прямое присваивание одной ссылке значения другой ссылки приведет к тому

***4***

29. Для представления символов в Java используется кодировка

30. Изменение отдельного символа строки

31. Получение отдельного символа строки

32. При именовании переменных в Java не допускаются

33. Строковые литералы в Java являются

34. Тип char относится к

35. У переменных типа String нет встроенных функций, поддерживающих следующую операцию над строками

36. Пул Java String это

37. Основное различие между StringBuffer и StringBuilder

38. StringBuffer является

***5***

39. Выберите метод, не преобразующий строку

40. Для какого потока не является необходимым вызов метода close

41. К стандартным потокам ввода-вывода Java не относится

42. Пакет java.lang является

43. Пакет java.util является

44. Строка не может быть преобразована в число типа double применением

45. Строка не может быть преобразована в число типа int применением

***6***

46. Глубокое копирование массива объектов может быть достигнуто

47. Глубокое копирование массива объектов — это…

48. К основному недостатку массива по сравнению с коллекцией относят

49. Коллекция — это…

50. Основные типы коллекций в Java не включают

***7***

51. В языке Java

52. Выберите оператор, не являющийся оператором цикла

53. Выберите оператор, являющийся оператором ветвления

54. Выберите оператор, являющийся оператором перехода

55. Оператор switch чаще всего используется

56. Скомпилируется ли следующий код: int i; int j; (false ? i: j) = 55;

57. Цикл for в форме for each используется

***8***

58. Выберите ложное высказывание относительно конструкторов в классах Java

59. Выберите ложное высказывание относительно конструкторов в классах Java

60. Для метода с одним параметром, которым является Java-объект, его поля будут передаваться

61. Методы с модификатором final

62. Перегрузка методов — это

63. Поле с модификатором final не может быть проинициализирована

64. При перегрузке методов

65. Статические методы и поля класса — это элементы…

***9***

66. Анонимный класс — это

67. Выберите ложное высказывание

68. Выберите ложное высказывание о вложенных классах

69. Для вызова метода суперкласса в классе-потомке (при наличии в классе-потомке одноименного метода) нужно

70. Для переопределения метода в классе потомке достаточно

71. Для указания того, что класс является подклассом используется

72. Если в суперклассе есть один конструктор с параметрами, а в классе-потомке определен новый конструктор, не содержащий вызов конструктора суперкласса, то

73. Ключевое слово super нельзя использовать

74. Переопределение методов при наследовании позволяет

***10***

75. В интерфейсе все поля являются по умолчанию

76. В интерфейсе все функции являются по умолчанию

77. Выберите ложное высказывание об абстрактных классах

78. Выберите ложное высказывание об интерфейсах

79. Динамическая диспетчеризация методов позволяет

80. Для расширения интерфейса используется ключевое слово

81. Для создания классов, которые не могут иметь потомков, применяется

82. Для указания того, что класс реализует интерфейс используется

83. Интерфейс — это

84. Класс с модификатором final

85. default-методы в интерфейсах это

86. Статические методы в интерфейсах это

***11***

87. Выберите ложное высказывание о перечислениях

88. Выберите ложное высказывание о перечислениях

89. Выберите неверное утверждение о пакетах

90. Ключ -classpath при запуске утилиты java используется для

91. Ключ -d при запуске утилиты javac используется для

92. Пакеты в Java подключаются с помощью директивы

93. Перечисление имеет предопределенный метод

94. Перечисление имеет предопределенный метод

***12***

95. Выберите ложное утверждение об исключениях

96. Выберите признак проверяемого исключения

97. Выход за пределы индексации массива относится к типу исключения

98. К классам исключений не относится

99. К основным методам класса Throwable не относится

100. Ошибка деления на ноль относится к классу исключения

101. Ошибка доступа к файлу относится к классу исключения

102. К проверяемым типам исключений относится

103. К непроверяемым типам исключений относится

***13***

104. Знак «\*» в регулярном выражении означает

105. Знак «.» в регулярном выражении означает

106. Квантификаторы нужны для

107. Круглые скобки в регулярном выражении служат для

108. Символьный класс определяет

109. Укажите, какая строки будут соответствовать указанному регулярному выражению [a-zA-Z]{1}[a-zA-Z\\d\\.\\\_]+@([a-zA-Z]+\\.){1,2}((net)|(com)|(org))

110. Символ «^» в регулярном выражении используется для

111. \D в регулярном выражении это

112. «Жадные» операторы это

113. Обратные ссылки используются для

114. Интернационализация - это

115. Локализация - это

116. Класс ResourceBundle используется для

117. Класс Locale используется для

118. Для форматирования чисел в Java используется класс

***14***

119. Восстановление объекта при сериализации производится

120. Выберите истинное утверждение о сериализации

121. Для указания того, что во время сериализации объекта некоторое поле нужно игнорировать, используется модификатор

122. К несериализуемому системному типу относится

123. К сериализуемым системным типам относится

124. Необходимым условием для сериализации объектов класса является

125. Сериализация — это

126. Сохранение объекта при сериализации производится

127. Десериализация объекта невозможна если

128. Сериализацию классов-наследников сериализуемого типа можно запретить, если

***15***

129. К методам, общим для всех объектов не относится

130. Функция clone()

131. Функция compareTo()

132. Функция equals()

133. Функция hashCode()

134. Функция toString()

135. Функция finalize()

***16***

136. Выберите истинное высказывание

137. Выберите корректный пример отношения композиции

138. Выберите ложное высказывание о спецификации абстрактного класса в UML

139. Выберите ложное высказывание о спецификации интерфейса в UML

140. Для обозначение статических атрибутов и операций в UML используется

141. Для обозначения абстрактного класса его имя и абстрактные функции

142. Для обозначения отношения обобщения в языке UML используется

143. Для обозначения реализации интерфейса в языке UML используется

144. Зависимость – это…

145. Постусловие операции это

146. Самый слабый тип отношений между классами это

***17***

147. Антипаттерн – это

148. Выберите метод, являющийся методом рефакторинга

149. К принципам Grasp не относится

150. Принцип "The Open-Closed Principle" гласит

151. Принцип "The Dependency Inversion Principle" гласит

152. Принцип "The Single Responsibility Principle" гласит

153. Принцип “The Liskov Substitution Principle” гласит

154. Принцип “The Interface Segregation Principle” гласит

155. Рефакторинг направлен на

***18***

156. К поведенческим паттернам относится паттерн

157. К порождающим паттернам проектирования не относится

158. К структурным паттернам относится паттерн

159. Паттерн Абстрактная Фабрика

160. Паттерн Адаптер

161. Паттерн Декоратор

162. Паттерн Итератор

163. Паттерн Команда

164. Паттерн Компоновщик

165. Паттерн Мост

166. Паттерн Одиночка

167. Паттерн Состояние

168. Паттерн Стратегия

169. Паттерн Строитель

170. Паттерн Фабрика

***19***

171. Выберите истинное высказывание относительно ограничений на обобщенные типы

172. Выберите ложное высказывание относительно ограничений на обобщенные типы

173. Метасимвольный аргумент

174. Обобщенные типы задаются с помощью

175. Ограничение на метасимвольный аргумент, заданное с помощью ключевого слова extends

176. Ограничение на метасимвольный аргумент, заданное с помощью ключевого слова super

177. При использовании generic типов

178. При использовании non-generic типов

***20***

179. Базовая концепция Java — Collection — является

180. В какую иерархию коллекций входит Map

181. Выберите операцию, которая не является общей для разных типов коллекций

182. Итератор — это

183. К коллекции, специально созданной для работы в многопоточном режиме, относится

184. Коллекция типа List — это

185. Коллекция типа Map — это

186. Коллекция типа Queue — это

187. Коллекция типа Set — это

188. Основные типы коллекций находятся в пакете

***21***

189. List в Java является

190. Интерфейс List не содержит описание следующего метода

191. К основным реализациям List не относится

192. Контейнер типа ArrayList представляет собой

193. Контейнер типа LinkedList представляет собой

194. Одна из особенностей ListIterator по сравнению с обычным Iterator в том

***22***

195. Map в Java является

196. Взаимодействие потоков может осуществляться с помощью функции

197. Выберите ложное утверждение о многопоточном программировании

198. Выберите ложное утверждение о множествах (Set)

199. Выберите ложное утверждение об отображениях (Map)

200. Главная особенность реализаций SortedSet в

201. Интерфейс Comparable предназначен для

202. Интерфейс Map не содержит описание следующего метода

203. Итерация по отображению

204. К основным реализациям Set не относится

205. К отображению, которое сохраняет элементы в отсортированном порядке относится

206. К преимуществам многопоточности не относится

207. К состояниям потока не относится

208. Компараторы позволяют

209. Почему Map не расширяет интерфейс Collection?

***23***

210. Test-Driven Development — это

211. Библиотека JUnit применяется для

212. Выберите истинное утверждение

213. Выберите ложное утверждение

214. Исчерпывающее тестирование (входное или тестирование путей)

215. Не существует стратегии тестирования методом

216. Тестирование – это

217. Тестирование методом белого ящика предполагает, что

218. Тестирование методом черного ящика предполагает, что

***24***

219. К методам тестирования "белого ящика" не относятся

220. К методам тестирования "черного ящика" не относится

221. Тестирование методом анализа граничных значений

222. Тестирование методом комбинаторного покрытия условий

223. Тестирование методом покрытия операторов

224. Тестирование методом покрытия решений

225. Тестирование методом покрытия условий

226. Тестирование методом эквивалентного разбиения

227. Причинно-следственные диаграммы

***25***

228. Инкрементное тестирование – это

229. К достоинствам восходящего тестирования относится

230. К достоинствам нисходящего тестирования относится

231. К недостаткам восходящего тестирования относится

232. К недостаткам нисходящего тестирования относится

233. Модульное тестирование – это

234. Неинкрементное тестирование – это

***26***

235. Категория "Возможности" в системном тесте предполагает

236. Категория "Восстанавливаемости" в системном тесте предполагает

237. Категория "Надежности" в системном тесте предполагает

238. Цель интеграционного теста

239. Цель модульного теста

240. Цель приемочного теста

241. Цель системного теста

242. Цель тестирования установки

243. Цель функционального теста

244. К критериям завершения тестов не относится

***28***

245. Объект, представляющий собой окно программы и содержащий все объекты JavaFX-приложения называется

246. Объект, представляющий физический контент JavaFX-приложения называется

247. Основным паттерном, используемым при разработке JavaFX-приложения, является

248. Разметка JavaFX-приложения хранится в файле с расширением

249. Макет JavaFX, размещающий все компоненты приложения последовательно друг на друге, называется

250. Макет JavaFX, добавляющий компоненты приложения в форме плиток одинакового размера, называется

251. Одновременные манипуляции, выполняемые несколькими потоками, над графом сцены JavaFX

252. Классы, применяемые для организации многопоточности в JavaFX, называются

***29***

253. Этот интерфейс применяется при использовании статических SQL-запросов, не изменяющихся в процессе работы

254. Этот интерфейс применяется, если SQL-запросы используют параметры, которые многократно изменяются в процессе работы

255. Метод Statement, возвращающий ResultSet-объект, называется

256. Интерфейс CallableStatement применяется для

257. С помощью этого интерфейса можно получить доступ к данным, полученным оператором SELECT

258. По выборке, получаемой с помощью объекта ResultSet, можно двигаться

259. Выборка, получаемая с помощью ResultSet является

260. Ключевая особенность транзакций -

261. Типы данных JDBC

262. Для закрепления изменений, произведенных транзакцией

263. Для включения режима транзакций при наличии объекта Connection conn используется

264. Для внесения изменений в таблицу базы данных

***30***

265. При использовании ссылок на методы можно ссылаться на

266. Функция является чистой, если

267. Функция имеет высший порядок, если

268.Функция не изменяет состояние, если

269.Функциональный интерфейс — это

270. Для сопоставления лямбда-выражения и интерфейса необходимо

271. К встроенным функциональным интерфейсам Java не относится

272. Функциональный интерфейс, представляющий функцию, принимающий один параметр и возвращающий единственное значение, это (выбрать наиболее подходящий)

273. Функциональный интерфейс, представляющий функцию, принимающий один параметр и возвращающий булево значение, это (выбрать наиболее подходящий)

274. Функциональный интерфейс, представляющий функцию, принимающий один параметр и возвращающий значение того же типа, это (выбрать наиболее подходящий)

275. Consumer — это функциональный интерфейс, представляющий функцию, которая

276. Техника, которая комбинирует множественные функции в единую функцию, которая использует все комбинируемые функции, это

277. Функциональная композиция предикатов может выполняться

278. Функциональная композиция функций может выполняться

279. Что такое Java Stream API

280. Нетерминальная потоковая операция это

281. Терминальная потоковая операция это

282. К терминальным потоковым операциям не относится метод

283. К терминальным потоковым операциям относится метод

284. К нетерминальным потоковым операциям не относится метод

285. К нетерминальным потоковым операциям относится метод

286. К недостаткам Java Stream API относится

287. Java Stream-объект может быть создан

288. Редукция Java Stream-объекта применяется