Образовательное частное учреждение
Дополнительного профессионального образования «Центр компьютерного обучения «Специалист» Учебно-научного центра при МГТУ им. Н.Э. Баумана»
(ОЧУ «Специалист»)

123317 Москва, Пресненская набережная, д 8, стр. 1, этаж 48, помещение 484с, комната 5
ИНН 7701257303, ОГРН 1037739408189

Утверждаю:
Директор ОЧУ «Специалист»

_____/Т.С.Григорьева/
«20» февраля 2018 года

Дополнительная профессиональная программа повышения квалификации «Java SE9. Уровень 2. Разработка клиент - серверных приложений»

город Москва

Программа разработана в соответствии с приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. N 499 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам".

Повышение квалификации слушателей, осуществляемое в соответствии с программой, проводится с использованием модульного принципа построения учебного плана с применением различных образовательных технологий, в том числе дистанционных образовательных технологий и электронного обучения в соответствии с законодательством об образовании.

Дополнительная профессиональная программа повышения квалификации, разработана образовательной организацией в соответствии с законодательством Российской Федерации, включает все модули, указанные в учебном плане.

Содержание оценочных и методических материалов определяется образовательной организацией самостоятельно с учетом положений законодательства об образовании Российской Федерации.

Структура дополнительной профессиональной программы соответствует требованиям Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденного приказом Минобрнауки России от 1 июля 2013 г. N 499.

Объем дополнительной профессиональной программы вне зависимости от применяемых образовательных технологий, должен быть не менее 16 академических часов. Сроки ее освоения определяются образовательной организацией самостоятельно.

Формы обучения слушателей (очная, очно-заочная, заочная) определяются образовательной организацией самостоятельно.

К освоению дополнительных профессиональных программ допускаются:

- лица, имеющие среднее профессиональное и (или) высшее образование;
- лица, получающие среднее профессиональное и (или) высшее образование.

Для определения структуры дополнительной профессиональной программы и трудоемкости ее освоения может применяться система зачетных единиц. Количество зачетных единиц по дополнительной профессиональной программе устанавливается организацией.

Образовательная деятельность слушателей предусматривает следующие виды учебных занятий и учебных работ: лекции, практические и семинарские занятия, лабораторные работы, круглые столы, мастер-классы, мастерские, деловые игры, ролевые игры, тренинги, семинары по обмену опытом, выездные занятия, консультации, выполнение аттестационной, дипломной, проектной работы и другие виды учебных занятий и учебных работ, определенные учебным планом.

1. Цель программы

Данный курс помогает слушателям углубить знания и навыки программирования на Java.

Совершенствуемые компетенции

№	Компетенция	Направление подготовки
		ФГОС ВО ПО НАПРАВЛЕНИЮ ПОДГОТОВКИ 09.03.02 «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»

		(УРОВЕНЬ БАКАЛАВРИАТА)
		Код компетенции
1	способностью участвовать в работах по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем	ПК-15
2	способность адаптировать приложения к изменяющимся условиям функционирования	ПК-32

Совершенствуемые компетенции в соответствии с трудовыми функциями профессионального стандарта «ПРОГРАММИСТ», утвержденного приказом Минтруда и социальной защиты РФ от 18 ноября 2013 г. N 679н

№	Компетенция	Направление подготовки
		Трудовые функции (код)
1	Разработка программного кода (Формализация и алгоритмизация поставленных задач, Написание программного кода с использованием языков программирования, определения и манипулирования данными, Оформление программного кода в соответствии с установленными требованиями)	A/01.3; A/02.3; A/03.3

Планируемый результат обучения

Лица, успешно освоившие программу, должны овладеть следующими компетенциями: научиться создавать современные приложения с многоуровневой архитектурой на Java Standard Edition (Java SE) и обеспечить эффективную работу этих приложений с использование многопоточного кода.

После окончания обучения слушатель будет знать:

- архитектуру для создания Java GUI-приложений
- основы web-приложений и взаимодействие с базами данных через JDBC API
- эффективность приложений, создание которых невозможно без многопоточного кода

После окончания обучения слушатель будет уметь:

- создавать многопоточные приложения;
- писать код синхронизации потоков;
- использовать пулы потоков;
- понимать архитектуру JavaFX;
- создавать современный интерфейс на JavaFX;
- реализовывать многопоточность в JavaFX;
- использовать JDBC в Java приложениях;
- работать с основными объектами JDBC API;
- использовать сервлеты и страницы JSP в распределённых приложениях Java;
- создавать приложения JavaServer Faces (JSF) ;
- разрабатывать пользовательские компоненты JSF;
- создавать и использовать веб-сервисы.

2. Учебный план

Категория слушателей:

- Программисты
- Разработчики клиент-серверных приложений
- Выпускники курса «Java SE9. Уровень 1. Основы программирования»

Требования к предварительной подготовке: успешное окончание курса «Java SE9. Уровень 1. Основы программирования».

Срок обучения: 40 академических часов с преподавателем.

Самостоятельные занятия: предусмотрены.

Форма обучения: очная, очно-заочная, заочная. По желанию слушателя форма обучения может быть изменена и/или дополнена.

Режим занятий: дневной, вечерний, группы выходного дня.

№ п/п	Наименование модулей по программе	Общая трудоемкость (акад. часов)	В том числе	
			Лекций	Практических занятий
1	Многопоточное программирование	10	5	5
2	Использование JDBC API в приложениях Java	10	5	5
3	Разработка современного пользовательского интерфейса на JavaFX	10	5	5
4	Web-приложения Java	10	5	5
	Итог:	40	20	20
	Итоговая аттестация	тестирование		

Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут.

3. Календарный учебный график

Календарный учебный график формируется при осуществлении обучения в течение всего календарного года. По мере набора групп слушателей по программе составляется календарный график, учитывающий объемы лекций, практики, самоподготовки, выезды на объекты.

Неделя обучения	1	2	3	4	5	6	7	Итого часов
	пн	вт	ср	чт	пт	сб	вс	
1 неделя	8	8	8	8	8 ИА	-	-	40
Итого:								40
Примечание: ИА – Итоговая аттестация (тестирование)								

4. Рабочие программы учебных предметов

Модуль 1. Многопоточное программирование

- Плюсы и минусы многопоточных приложений
- Средства Java для управления многопоточностью
- Класс Thread и интерфейс Runnable
- Создание потоков
- Мониторы и синхронизация потоков
- Современные средства по управлению потоками (Executors, Fork/Join Framework)
- Новые потоки, безопасные коллекции и классы (ThreadLocalRandom, AtomicInteger и др.)
- Reactive streams (Java SE9)

Лабораторная работа. Создание и синхронизация потоков.

Модуль 2. Использование JDBC API в приложениях Java

- Java и взаимодействие с СУБД
- JDBC, использование SQL в Java-приложениях для доступа к реляционным БД
- JDBC-драйвера, их виды
- Основные объекты JDBC
- Транзакции JDBC

Лабораторная работа. Создание приложения Java/JDBC для работы с изображениями.

Модуль 3. Разработка современного пользовательского интерфейса на JavaFX

- История GUI.
- Обзор возможностей JavaFX.
- Основные объекты в архитектуре JavaFX.
- Интеграция JavaFX и Swing.
- Создание простого приложения JavaFX и JavaFX FXML.
- Коллекции JavaFX.

Лабораторная работа. Использование коллекций в пользовательских интерфейса JavaFX.

- Работа с элементами управления и событиями в JavaFX.
- Контейнера JavaFX.
- Использование CSS.
- Создание диаграмм и WebView.
- Визуальные эффекты и анимация в JavaFX
- Использование свойств и привязки данных в JavaFX.
- Реализация многопоточности в JavaFX.

Лабораторная работа. Создание современного пользовательского интерфейса с JavaFX.

Модуль 4. Web-приложения Java

- Архитектура распределенных приложений.
- Сервлеты и страницы JSP.
- Введение в JavaServer Faces (JSF).
- Структура JSF приложения.
- Компоненты ввода-вывода в JSF.
- Создание пользовательских компонентов в JSF.
- Веб-сервисы.

Лабораторная работа.

- Создание приложения с использованием JSP.
- Использование веб-сервисов.

5. Организационно- педагогические условия

Требования к кадровым условиям реализации дополнительной профессиональной программы:

а) преподавательский состав образовательной организации, обеспечивающий образовательный процесс, должен обладать высшим образованием и стажем преподавания по изучаемой тематике не менее 1 года и (или) практической работы в областях знаний, предусмотренных модулями программы, не менее 3 (трех) лет;

б) образовательной организацией наряду с традиционными лекционно-семинарскими занятиями должны применяться современные эффективные методики преподавания с применением интерактивных форм обучения, аудиовизуальных средств, информационно-телекоммуникационных ресурсов и наглядных учебных пособий.

Требования к материально-техническому и учебно-методическому обеспечению дополнительной профессиональной программы:

а) образовательная организация располагает необходимой материально-технической базой, включая современные аудитории, библиотеку, аудиовизуальные средства обучения, мультимедийную аппаратуру, оргтехнику, копировальные аппараты. Материальная база соответствует санитарным и техническим нормам и правилам и обеспечивает проведение всех видов практической и дисциплинарной подготовки слушателей, предусмотренных учебным планом реализуемой дополнительной профессиональной программы.

б) в случае применения электронного обучения, дистанционных образовательных технологий каждый обучающийся в течение всего периода обучения должен быть обеспечен индивидуальным неограниченным доступом к электронной информационно-образовательной среде, содержащей все электронные образовательные ресурсы, перечисленные в модулях дополнительной профессиональной программы.

6. Формы аттестации и оценочные материалы

Образовательная организация несет ответственность за качество подготовки слушателей и реализацию дополнительной профессиональной программы в полном объеме в соответствии с учебным планом.

Оценка качества освоения дополнительной профессиональной программы слушателей включает текущий контроль успеваемости и итоговую аттестацию.

Конкретные формы и процедуры текущего контроля успеваемости, промежуточной аттестации и итоговой аттестации слушателей устанавливаются образовательной организацией самостоятельно.

Слушателям, успешно освоившим дополнительную профессиональную программу и прошедшим итоговую аттестацию, выдается удостоверение о повышении квалификации.

Слушателям, не прошедшим итоговой аттестации или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть дополнительной профессиональной программы и (или) отчисленным из образовательной организации, выдается справка об обучении или о периоде обучения по образцу, самостоятельно устанавливаемому образовательной организацией.

Итоговая аттестация проводится по форме тестирования в соответствии с учебным планом.

Вопрос 1

Какие результаты можно увидеть на экране при попытке компиляции и запуска следующей программы?

final class My{
 public static void main(String a){
 System.out.println("Hello!");
 }
}

Выберите один ответ:

- Hello!
- Программа не запустится из-за отсутствия метода main
- Программа не откомпилируется из-за недопустимого спецификатора доступа в описании класса
- Программа не откомпилируется из-за того, что компилятор не найдет классы System и String

Вопрос 2

Какие результаты можно увидеть на экране при попытке компиляции и запуска следующей программы?

class First{
 int sum(int x,int y){
 return(x+y);
 }
}
public class Second extends First{
 public static void main(String a[]){
 Second s=new Second();
 System.out.println("Summa="+s.sum(5,6));
 }
}

Выберите один ответ:

- Программа не откомпилируется из-за отсутствия спецификатора доступа public в описании класса First
- Программа не откомпилируется из-за невозможности создать объект экземпляр класса Second внутри класса Second
- Summa=11
- Программа не откомпилируется из-за того что метод sum не унаследуется классом Second

Вопрос 3

Какие результаты можно увидеть на экране при попытке компиляции и запуска следующей программы?

interface First{
 int a=5;
}
public class Second implements First{
 public static void main(String arg[]){
 a=6;
 System.out.println("a="+a);
 }
}

Выберите один ответ:

- Программа не откомпилируется из-за отсутствия спецификатора доступа public в описании интерфейса First
- Программа не откомпилируется из-за отсутствия спецификатора доступа public в описании поля a внутри интерфейса First
- Программа не откомпилируется из-за недопустимости описания полей внутри интерфейсного класса

- Программа не откомпилируется из-за недопустимости присваивания значений полям интерфейсного класса
- Программа не откомпилируется из-за недопустимости использования нестатических полей внутри статического метода
- Программа не откомпилируется из-за недопустимости присваивания нового значения полю a, так как это поле имеет спецификатор доступа final

Вопрос 4

Какие определения метода sum являются правильными?

A. public class My{
private int sum(int x,int y){
 return(x+y);
}
}
B. class My{
public int sum(int x,int y){
 return(x+y);
}
}
C. class My{
public final int sum(int x,int y){
 return(x+y);
}
}
D. class My{
public int sum(int x,int y);
}
E. abstract class My{
 abstract public int sum(int x,int y);
}
abstract class My{
abstract public final int sum(int x,int y);
}

Выберите несколько ответов:

- A
- B
- C
- D
- E
- F

Вопрос 5

Какие определения интерфейса My являются правильными?

A. interface My{
 public int sum(int x,int y){
 return(x+y);
 }
}
B. interface My{
 int sum(int x,int y);
}
C. public interface My{
 public int sum(int x,int y);
}
D. public interface My{
 public abstract int sum(int x,int y);
}
E. public interface My{
 public int a;
 public int sum(int x,int y);
}
F. public interface My{
 public int a=0;
 public int sum(int x,int y);
}

Выберите несколько ответов:

- A
- B
- C
- D
- E
- F

Вопрос 6

Какие результаты можно увидеть на экране при попытке компиляции и запуска следующей программы?

public class My{
 public static void main(String arg[]){
 int x=0, y;
 if(x==0)y=1;
 if(x!=0)y=2;
 System.out.println("y="+y);
 }
}

Выберите один ответ:

- программа откомпилируется и при выполнении вернет 0
- программа откомпилируется и при выполнении вернет 1
- программа откомпилируется и при выполнении вернет null
- программа откомпилируется и при выполнении вернет 2
- программа не откомпилируется, так как переменная y может быть не инициализирована
- программа откомпилируется и при выполнении вернет сообщение о ошибке

Вопрос 7

Какие результаты можно увидеть на экране при попытке компиляции и запуска следующей программы?

public class My{
 static int x,y;
 public static void main(String arg[]){
 if(x==0)
 y=1;
 else
 y=2;
 System.out.println("y="+y);
 }
}

Выберите один ответ:

- программа откомпилируется и при выполнении вернет 0
- программа откомпилируется и при выполнении вернет 1
- программа откомпилируется и при выполнении вернет null
- программа откомпилируется и при выполнении вернет 2
- программа не откомпилируется, так как переменная y может быть не инициализирована
- программа откомпилируется и при выполнении вернет сообщение о ошибке

Вопрос 8

Какие результаты можно увидеть на экране при попытке компиляции и запуска следующей программы?

public class My{
 int x=5, y=6;
 class MyToo{
 int sum(){
 return(My.this.x+My.this.y);
 }
 }
 public static void main(String arg[]){
 My m=new My();
 My.MyToo mt=m.new MyToo();
 System.out.println("sum="+mt.sum());
 }
}

Выберите один ответ:

- программа откомпилируется и при выполнении вернет null
- программа откомпилируется и при выполнении вернет 11

- программа не откомпилируется, так как поля x и y не доступны внутри класса MyToo
- программа не откомпилируется, так как неправильно создается объект экземпляр класса MyToo
- программа не откомпилируется, так как методы вложенного класса не имеют спецификатора доступа public
- программа не откомпилируется, так как метод sum вложенного класса должен иметь спецификатор доступа static
- программа откомпилируется и при выполнении вернет сообщение о ошибке

Вопрос 9

Какой класс является родительским для потоков ввода данных?

Выберите один ответ:

- InputStream
- OutputStream
- FileInputStream
- DataStream

Вопрос 10

Какой класс является родительским для потоков вывода данных?

Выберите один ответ:

- InputStream
- OutputStream
- FileOutputStream
- DataStream

Вопрос 11

Экземпляром какого класса является поле System.err?

Выберите один ответ:

- java.lang.System
- java.io.OutputStream
- java.io.ErrorStream
- java.io.PrintStream

Вопрос 12

Экземпляром какого класса является поле System.out?

Выберите один ответ:

- java.lang.System
- java.io.OutputStream
- java.io.DataOutputStream
- java.io.PrintStream
- java.io.OutputStreamWriter

Вопрос 13

Экземпляром какого класса является поле System.in?

Выберите один ответ:

- java.lang.System
- java.io.InputStream
- java.io.BufferedInputStream
- java.io.PrintStream
- java.io.InputStreamWriter

Вопрос 14

Какой класс пакета java.io позволяет создать поток ввода текстовых данных с учетом кодировки?

Выберите один ответ:

- Reader
- InputStream
- InputStreamReader
- FilterInputStream
- CharArrayReader

Вопрос 15

Какой класс пакета java.io позволяет создать поток вывода текстовых данных с учетом кодировки?

Выберите один ответ:

- Writer
- OutputStream
- OutputStreamWriter
- FilterOutputStream
- CharArrayWriter

Вопрос 16

Какая ошибка возникнет при компиляции следующего фрагмента кода?

import java.io.*;
public class My{
 public static void main(String arg[]){
 byte b[]=new byte[256];
 FileInputStream fis=new FileInputStream("test.txt");
 fis.read(b);
 fis.close();
 }
}

Выберите один ответ:

- Никакой ошибки не будет
- Ошибка возникнет из-за отсутствия метода read в классе FileInputStream
- Ошибка возникнет из-за того, что конструктор класса FileInputStream и методы read и close могут порождать исключительную ситуацию типа IOException
- Ошибка возникнет из-за необходимости указывать полный путь к файлу в конструкторе класса FileInputStream
- Ошибка возникнет из-за того, что длина массива b может не совпадать с длиной файла "test.txt"
- Ошибка возникнет из-за того, что нельзя читать данные в байтовый массив из текстового файла

Вопрос 17

Какая ошибка возникнет при компиляции следующего фрагмента кода?

public class My{
 public static void main(String arg[]){
 byte b[]=new byte[256];
 System.in.read(b);
 }
}

Выберите один ответ:

- Никакой ошибки не будет
- Ошибка возникнет из-за того, что метод read может порождать исключительную ситуацию типа IOException
- Ошибка возникнет из-за того, что длина массива b может не совпадать с длиной вводимых данных
- Ошибка возникнет из-за того, что вводимые данные никуда не считываются

Вопрос 18

Какое сообщение возникнет при компиляции следующего фрагмента кода?

import java.io.*;
public class My{
 public static void main(String arg[]){
 String s;
 try{
 DataInputStream dis=new DataInputStream(new FileInputStream("test.txt"));
 s=dis.readLine();
 dis.close();
 }catch(IOException e){}
 }
}

Выберите один ответ:

- Никаких сообщений не будет
- Возникнет сообщение о ошибке, так как в конструкторе класса DataInputStream должен быть аргумент типа InputStream
- Возникнет сообщение о ошибке, так как конструктор класса FileInputStream может порождать исключительную ситуацию типа FileNotFoundException
- Возникнет сообщение о том, что в классе My используются запрещенные методы

Вопрос 19

Какие классы являются наследниками от класса java.awt.Container?

Выберите несколько ответов:

- java.awt.Component
- java.awt.Panel
- java.applet.Applet
- java.awt.Frame
- java.awt.Dialog

Вопрос 20

Какой менеджер компоновок позволяет разместить компоненты в таблице с ячейками равного размера?

Выберите один ответ:

- FlowLayout
- GridLayout
- BorderLayout
- CardLayout

Вопрос 21

Что будет видно в окне, созданном при помощи приведенного кода?

import java.awt.*;
public class My{
 public static void main(String arg[]){
 Frame f=new Frame();
 f.setLayout(new CardLayout());
 f.add(new Button("A"),"First");

f.add(new Button("B"),"Second");
 f.add(new Button("C"),"Thead");
 f.setSize(200,200);
 f.setVisible(true);
 }
}

Выберите один ответ:

- Кнопка "A"
- Кнопка "C"
- Кнопки "A","B","C" в одну строчку
- Кнопки "A","B","C" в один столбик
- Ничего в окне не появится

Вопрос 22

Что будет видно в окне, созданном при помощи нижеприведенного кода?

import java.awt.*;
public class My{
 public static void main(String arg[]){
 Frame f=new Frame();
 f.setLayout(new GridLayout(1,3,3,1));
 f.add(new Button("A"),"First");
 f.add(new Button("B"),"Second");
 f.add(new Button("C"),"Thead");
 f.setSize(400,400);
 f.setVisible(true);
 }
}

Выберите один ответ:

- Кнопка "A"
- Кнопка "C"
- Кнопки "A","B","C" в одну строчку
- Кнопки "A","B","C" в один столбик
- Ничего в окне не появится

Вопрос 23

Какое выравнивание используется по умолчанию для компоновки FlowLayout?

Выберите один ответ:

- FlowLayout.RIGHT
- FlowLayout.LEFT
- FlowLayout.CENTER
- FlowLayout.MIDDLE
- Выравнивание надо всегда указывать явно

Вопрос 24

Какие результаты можно увидеть в окне при попытке компиляции и запуска следующей программы?

import java.awt.*;
public class My extends Frame{
 public void paint(Graphics g){
 Font f=new Font(“Gothic”,Font.BOLD,24);
 g.setFont(f);

g.drawString(“Hello”,20,20);
 }
 public static void main(String arg[]){
 My m=new My();
 m.setSize(200,200);
 m.setVisible(true);
 }
}

Выберите один ответ:

- программа не откомпилируется, так как шрифт Gothic не определен в Java
- программа откомпилируется и в окне появится слово Hello выведенное шрифтом Gothic
- программа откомпилируется и в окне появится слово Hello выведенное шрифтом по умолчанию
- программа откомпилируется, но при попытке запуска возникнет исключительная ситуация типа NullPointerException

Вопрос 25

Каким требованиям должен обязательно удовлетворять класс чтобы являться апплетом?

Выберите один ответ:

- Он должен быть наследником класса Applet и у него обязательно не должно быть метода main
- Он должен быть наследником класса Applet или его потомка
- Он должен быть наследником класса Applet и в нем необходимо переопределить унаследованный от класса Applet метод paint
- Он должен быть наследником класса Applet и и нем необходимо переопределить унаследованный от класса Applet метод init
- Он должен быть наследником класса Applet и все его методы должны иметь спецификатор доступа public
- Он должен быть наследником класса Applet и у него должен быть конструктор по умолчанию

Вопрос 26

Какой тэг необходимо использовать для размещения апплета в HTML-странице?

Выберите один ответ:

- EMBED
- INCLUDE
- APPLET
- CODE
- PARAM

Вопрос 27

Какой атрибут тэга APPLET обязателен?

Выберите один ответ:

- CODE
- CODEBASE
- ARCHIVE
- URL
- SRC

Вопрос 28

Какие утверждения верны?

Выберите несколько ответов:

- Апплеты не могут создавать сетевые соединения ни с каким компьютером
- Апплеты не могут создавать сетевые соединения, за исключением того компьютера с которого загружены
- Апплеты не имеют доступа к файловой системе клиентского компьютера
- Апплеты не могут взаимодействовать с базами данных
- Апплеты не могут быть многопоточными
- Апплет это полноценная Java-программа не имеющая никаких принципиальных ограничений

Вопрос 29

Сколько типов драйверов баз данных различают в JDBC?

Выберите один ответ:

- Никаких различий в типах драйверов не существует
- 2
- 3
- 4

Вопрос 30

Может ли апплет осуществить соединение с базой данных, находящейся на том же сервере, откуда скачан апплет?

Выберите один ответ:

- Да, при наличии JDBC драйвера 4-типа

- Да, при наличии JDBC драйвера 1-типа
- Нет, из-за ограничений по безопасности
- Нет, так как осуществлять соединения с базой данной могут только консольные приложения

Вопрос 31

Какой пакет необходимо использовать для работы с базами данных?

Выберите один ответ:

- java.sql
- java.jdbc
- javax.sql
- java.db
- sun.jdbc

Вопрос 32

Какой метод класса Statement надо использовать для посылки SQL-команды "INSERT INTO Table (X,Y) VALUES (5,6)"?

Выберите один ответ:

- executeSQL
- executeQuery
- executeUpdate
- executeInsert

Вопрос 33

Какой метод класса Statement надо использовать для посылки SQL-команды "UPDATE Table SET X=5,Y=6"?

Выберите один ответ:

- executeSQL
- executeQuery
- executeUpdate
- executeSQLUpdate

Вопрос 34

Какой метод класса Statement надо использовать для посылки SQL-команды "SELECT * FROM Table"?

Выберите один ответ:

- executeSQL
- executeQuery
- executeUpdate
- executeSelect

Вопрос 35

Исключительная ситуация какого типа возможна при использовании метода execute класса Statement

Выберите один ответ:

- DBException
- SQLException
- DataBaseException
- BadResultException
- JDBCException
- Никакой исключительной ситуации не предусмотрено

Вопрос 36

Какой класс необходимо использовать для открытия серверного сокета?

Выберите один ответ:

- Socket
- Server
- ServerSocket
- SocketServer

Вопрос 37

Какой класс необходимо использовать клиенту для соединения с сокетом сервера?

Выберите один ответ:

- Socket
- Server

- ServerSocket
- SocketServer

Вопрос 38

Какой метод использует класс ServerSocket для "прослушивания" своего сокета?

Выберите один ответ:

- listen
- accept
- hear
- getClientSocket
- getClient

Вопрос 39

Исключительная ситуация какого типа возможна при открытии сокетного соединения следующим образом: Socket s=new Socket("www.specialist.ru",80);

Выберите один ответ:

- только IOException
- только MalformedURLException
- IOException и MalformedURLException
- только UnknownHostException
- IOException и UnknownHostException
- только BadURLException

Вопрос 40

Каким требованиям должен обязательно удовлетворять класс чтобы являться сервлетом?

Выберите один ответ:

- Он должен реализовывать интерфейс Servlet и у него обязательно не должно быть метода main
- Он должен реализовывать интерфейс Servlet
- Он должен быть наследником класса HttpServlet и в нем обязательно необходимо переопределить унаследованные методы doGet и doPost
- Он должен реализовывать интерфейс Servlet и в нем необходимо переопределить унаследованный метод init

- Он должен реализовывать интерфейс Servlet и все его методы должны иметь спецификатор доступа public
- Он должен реализовывать интерфейс Servlet и у него должен быть конструктор по умолчанию

Вопрос 41

В каком пакете находится интерфейс Servlet?

Выберите один ответ:

- В пакете java.servlet
- В пакете javax.servlet
- В пакете javax.servlet.http
- В пакете java.http

Вопрос 42

Что увидит пользователь в результате просмотра следующей jsp-страницы?

<HTML>
<BODY>
<%-- String s=”Hello” --%>
<%! String s=”Hello JSP” %>
<H1 ALIGN=”center”><%=s%></H1>
</BODY>
</HTML>

Выберите один ответ:

- Пустую страницу
- Сообщение об ошибке
- Hello
- Hello JSP
- null

Вопрос 43

Для просмотра jsp-страницы с именем My.jsp пользователь набрал в адресной строке броузера следующий адрес: http://hostname/My.jsp?f1=Hello ,что увидит он в результате просмотра этой страницы, если путь указан верно, а код страницы приведен ниже?

<HTML>
<BODY>
<%! String s=”Hello JSP” %>
<%s=request.getParameter("f1")%>
<H1 ALIGN=”center”><%=s%></H1>
</BODY>
</HTML>

Выберите один ответ:

- Пустую страницу
- Сообщение об ошибке
- Hello
- Hello JSP
- null