1. **Паспорт Образовательной программы**

**«Программирование на языке Java. Базовый курс»**

|  |  |
| --- | --- |
| **Версия программы** | 7 |
| **Дата Версии** | 01.09**.**2020 |

1. **Сведения о Провайдере**

|  |  |  |
| --- | --- | --- |
| 1.1 | Провайдер | Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» |
| 1.2 | Логотип образовательной организации |  |
| 1.3 | Провайдер ИНН | 7021000043 |
| 1.4 | Ответственный за программу ФИО | Ковшов Алексей Валерьевич |
| 1.5 | Ответственный должность | Начальник управления дополнительного образования Института инноватики |
| 1.6 | Ответственный Телефон | +7 3822 701736 |
| 1.7 | Ответственный Е-mail | do@2i.tusur.ru |

1. **Основные Данные**

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Описание** |
| 2.1 | Название программы | Программирование на языке Java. Базовый курс |
| 2.2 | Ссылка на страницу программы | <https://do.tusur.ru/courses/programs/java> |
| 2.3 | Формат обучения | Онлайн |
| 2.4 | Подтверждение от ОО наличия возможности реализации образовательной программы с применением электронного обучения и (или) дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа | ТУСУР проводит обучение с применением технологий дистанционного/электронного обучения с 1998 года. Дополнительные образовательные программы реализуются в собственной системе онлайн-обучения по адресу <https://udo.tusur.ru/>, доступ - только для обучающихся |
| 2.5 | Уровень сложности | Базовый |
| 2.6 | Количество академических часов | **72** |
| 2.7 | Практикоориентированный характер образовательной программы: не менее 50 % трудоёмкости учебной деятельности отведено практическим занятиям и (или) выполнению практических заданий в режиме самостоятельной работы (кол-во академических часов) | 49 академических часов отводится на выполнение практических задний |
| 2.8 | Стоимость обучения одного обучающегося по образовательной программе, а также предоставление ссылок на 3 (три) аналогичные образовательные программы иных организаций, осуществляющих обучение, для оценки объективности стоимости или обоснование уникальности представленной образовательной программы в случае отсутствия аналогичных образовательных программ на рынке образовательных услуг | Стоимость обучения - 20 000 руб. за одного человека  Аналогичные образовательные программы:  1. "Разработчик Java. Базовый уровень" (72 ч.). Академия АйТи. Стоимость обучения - 22 800 руб. (<https://www.academyit.ru/courses/J100/>)  2. "Язык программирования Java (Java SE10). Уровень 1. Базовый курс" (40 ч.). МГТУ им. Н.Э. Баумана. Стоимость обучения - 24 950 руб. (http://edu.bmstu.ru/napravleniya-obucheniya/programmirovanie/java1)  3. "Основы программирования на Java. Уровень 1" (40 ч.). Центр IT-Образования Level UP. Стоимость обучения – 14 500 руб. (http://levelp.ru/courses/programmirovanie/basics-of-programming-in-the-java-language-level-1/) |
| 2.9 | Минимальное количество человек на курсе | 1 |
| 2.10 | Максимальное количество человек на курсе | До 30 одновременно |
| 2.11 | Данные о количестве слушателей, ранее успешно прошедших обучение по образовательной программе | 300+ |
| 2.12 | Формы аттестации | зачет |
| 2.13 | Указание на область реализации компетенций цифровой экономики, к которой в большей степени относится образовательная программа, в соответствии с Перечнем областей | Программирование и создание ИТ-продуктов |

1. **Аннотация программы**

Наиболее полное и содержательное описание программы, которое включает:

1) общую характеристику компетенций, качественное изменение которых осуществляется в результате обучения или которые формируются в результате освоения образовательной программы;

2) описание требований и рекомендаций для обучения по образовательной программе;

3) краткое описание результатов обучения в свободной форме, а также описание востребованности результатов обучения в профессиональной деятельности.

Ограничение по размеру: не менее 1000 символов -?

В современной IT-индустрии язык Java является одним из самых востребованных объектно-ориентированных языков программирования. Java стремительно развивается и охватывает практически все области программирования. Java используется для создания серверного программного обеспечения, прикладных программ, разработки веб-сайтов, игр и многого другого. Приложения, написанные на Java, могут работать на любом компьютере независимо от его архитектуры.

Java-разработчик должен обладать уверенными знаниями объектно-ориентированного программирования, уметь писать простой читабельный код с применением шаблонов проектирования. Java имеет множество библиотек и фреймворков. Разработчикам нужно знать их не только на уровне использования, но и понимать, как эти библиотеки и фреймворки устроены внутри.

Java-программисты востребованы в тех областях, где используют проекты с многопользовательской обработкой данных и высокими требованиями к уровню безопасности. Java-программист обычно разрабатывает архитектуру приложения, бизнес-логику приложения, поддерживает и сопровождает проект и пишет техническую документацию.

Программа предназначена для:

– студентов и выпускников вузов и колледжей любых профилей, желающих получить новые знания и навыки, востребованные в IT-индустрии,

– специалистов, желающих повысить свою квалификацию в области технологий разработки программного обеспечения.

– всех желающих начать карьеру Java-разработчика.

Цель реализации программы – способность применять основы языка программирования Java, необходимые для успешного освоения любой из высокоуровневых технологий семейства Java.

Федеральное государственное бюджетное  
образовательное учреждение высшего образования   
«Томский государственный университет  
систем управления и радиоэлектроники»

Дополнительная профессиональная программа повышения квалификации

«Программирование на языке Java. Базовый курс »

72 час.

**ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ**

**1.Цель программы**

Формирование способности применять основы языка программирования Java, необходимые для успешного освоения любой из высокоуровневых технологий семейства Java

**2.Планируемые результаты обучения:**

2.1.Знание (осведомленность в областях)

2.1.1. истории возникновения и развития технологии Java, основных целей и областей применения, основных понятий и компонент технологии Java, состава комплекта разработки JDK, редакций и версий Java, жизненного цикла Java-программы;

2.1.2. основных элементов интерфейса и возможностей среды разработки Eclipse, возможностей, появившихся в Eclipse для интеграции новых возможностей Java;

2.1.3. базовых понятий языка Java: класс, объект, атрибут, операция, композиция, наследование, интерфейс, пакет;

2.1.4. синтаксиса языка Java: идентификаторы, типы данных, массивы примитивных типов, переменные, арифметические и логические операторы и выражения;

2.1.5. основных возможностей класса String, классов Java API, массивов объектов

2.2. Умение (способность к деятельности)

2.2.1. устанавливать комплект разработки JDK и среду разработки Eclipse в операционной системе Windows или Linux;

2.2.2. компилировать и запускать программы на языке Java из командной строки;

2.2.3. манипулировать ресурсами в среде разработки Eclipse;

2.2.4. проектировать классы и применять механизмы наследования и полиморфизма;

2.2.5. выражать отношения между классами с помощью нотации UML;

2.2.6. использовать механизм исключений при операциях ввода-вывода

2.3.Навыки (использование конкретных инструментов)

2.3.1. разработки и отлаживания кода Java в среде разработки Eclipse;

2.3.2. проектирования графического интерфейса пользователя с помощью библиотеки Swing

**3.Категория слушателей** (возможно заполнение не всех полей)

* 1. Граждане старше 18 лет, не достигшие пенсионного возраста, имеющие среднее профессиональное и/или высшее образование
  2. Для указанных лиц обучение по программе будет считаться повышением квалификации с получением удостоверения установленного образца.
  3. Наличие опыта профессиональной деятельности: не требуется
  4. Предварительное освоение иных дисциплин/курсов /модулей: не требуется

**4.Учебный план программы «**Программирование на языке Java. Базовый курс**»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль** | **Всего, час** | **Виды учебных занятий** | | |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Тема 1. Технология Java и среда разработки JetBrains IDEA | 7 | 3 | 2 | 2 |
| 2 | Тема 2. Введение в объектно-ориентированное программирование на Java | 5 | 2 | 2 | 1 |
| 3 | Тема 3. Типы данных, переменные, операторы | 5 | 2 | 2 | 1 |
| 4 | Тема 4. Создание и использование объектов. Классы и объекты String. Классы и объекты Java API | 5 | 2 | 2 | 1 |
| 5 | Тема 5. Операторы управления ходом программы | 7 | 2 | 3 | 2 |
| 6 | Тема 6. Разработка методов, инкапсуляция и конструкторы, массивы объектов | 13 | 4 | 5 | 4 |
| 7 | Тема 7. Наследование, интерфейсы и полиморфизм | 15 | 4 | 7 | 4 |
| 8 | Тема 8. Механизм исключений и операции ввода/вывода | 6 | 2 | 2 | 2 |
| 9 | Тема 9. Коллекции, стримы, лямбда | 9 | 2 | 4 | 3 |
| **Итоговая аттестация** | |  | **Указывается вид (экзамен, зачёт, реферат и т.д.)** | | |
|  | | 72 | зачет | | |

**5.Календарный план-график реализации образовательной** программы

(дата начала обучения – дата завершения обучения) в текущем календарном году, указания на периодичность набора групп (не менее 1 группы в месяц)

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование учебных модулей** | **Трудоёмкость (час)** | **Сроки обучения** |
| **1** | Технология Java и среда разработки JetBrains IDEA | 8 | 01.11.20-02.11.20 |
| **2** | Введение в объектно-ориентированное программирование на Java | 8 | 03.11.20-05.11.20 |
| 3 | Типы данных, переменные, операторы | 8 | 06.11.20-08.11.20 |
| 4 | Создание и использование объектов. Классы и объекты String. Классы и объекты Java API | 8 | 09.11.20-11.11.20 |
| 5 | Операторы управления ходом программы | 8 | 12.11.20-14.11.20 |
| 6 | Разработка методов, инкапсуляция и конструкторы, массивы объектов | 8 | 15.11.20-17.11.20 |
| 7 | Наследование, интерфейсы и полиморфизм | 8 | 18.11.20-20.11.20 |
| 8 | Механизм исключений и операции ввода/вывода | 8 | 21.11.20-23.11.20 |
| 9 | Коллекции, стримы, лямбда | 8 | 24.11.20-25.11.20 |
| **Всего:** | | 72 | 01.11.20-25.11.20 |

**6.Учебно-тематический план программы «**  Программирование на языке Java. Базовый курс **»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль / Тема** | **Всего, час** | **Виды учебных занятий** | | | **Формы контроля** |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Технология Java и среда разработки JetBrains IDEA | 7 | 3 | 2 | 2 |  |
| 2 | Введение в объектно-ориентированное программирование на Java | 5 | 2 | 2 | 1 |  |
| 3 | Типы данных, переменные, операторы | 5 | 2 | 2 | 1 |  |
| 4 | Создание и использование объектов. Классы и объекты String. Классы и объекты Java API | 5 | 2 | 2 | 1 |  |
| 5 | Операторы управления ходом программы | 7 | 2 | 3 | 2 |  |
| 6 | Разработка методов, инкапсуляция и конструкторы, массивы объектов | 13 | 4 | 5 | 4 |  |
| 7 | Наследование, интерфейсы и полиморфизм | 15 | 4 | 7 | 4 |  |
| 8 | Механизм исключений и операции ввода/вывода | 6 | 2 | 2 | 2 |  |
| 9 | Коллекции, стримы, лямбда | 9 | 2 | 4 | 3 |  |
|  | Итоговая аттестация |  |  |  |  | зачет |

**7. Учебная (рабочая) программа повышения квалификации «**Программирование на языке Java. Базовый курс  **»**

**Тема 1.** Технология Java и среда разработки JetBrains IDEA **(** 7 **час)**

Краткая история развития. Основные понятия Java. Java Software Development Kit (JDK). Редакции и нумерация версий Java. Эволюция Java. Жизненный цикл программы на языке Java. Компиляция и запуск приложений. Запуск Java-приложений с помощью интерпретатора java из командной строки. Приложения на Java

**Тема 2**.Введение в объектно-ориентированное программирование на Java (5 час)

Предварительная подготовка, определения ООП, примеры, выводы. Что такое Объект? Что такое Класс? Что такое Наследование? Что такое Интерфейс? Что такое Пакет?

Тема 3**.** Типы данных, переменные, операторы **(**5 **час.)**

Идентификаторы. Типы данных и литералы. Бинарные литералы и использование символа подчеркивания в литералах. Приведение типов. Объявление и инициализация переменных. Соглашения о наименовании (Naming convention). Массивы примитивных типов. Операторы.

**Тема** 4**.** Создание и использование объектов. Классы и объекты String. Классы и объекты Java API (5 час.)

Объявление, создание экземпляров и инициализация. Хранение ссылочных переменных и объектов в оперативной памяти. Использование класса String. Создание строковых переменных. Хранение строк в памяти. Спецификация классов Java API.

Тема 5. Операторы управления ходом программы (7 час.)

Оператор цикла while. Оператор цикла do-while. Оператор цикла for, и оператор запятая. Операторы break и continue. Имеющий дурную славу “goto”. Операторы условия if-else и возврата return. Оператор выбора switch.

Тема 6. Разработка методов, инкапсуляция и конструкторы, массивы объектов (13 час.)

Разработка методов в классах Java. Инкапсуляция и конструкторы. Массивы и объекты.

Тема 7. Наследование, интерфейсы и полиморфизм (15 час.)

Наследование. Интерфейсы и абстрактные классы. Полиморфизм. Ключевое слово static. Ключевое слово final.

Тема 8. Механизм исключений и операции ввода/вывода (6 час.)

Механизм исключений. Файловый ввод\вывод. Сетевое взаимодействие. Нововведения Java 1.7: multi-catch, try-with-resources, выброс исключения с более точным указанием типа.

Тема 9. Коллекции, стримы, лямбда (9 час.)

**Описание практико-ориентированных заданий и кейсов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Номер темы/модуля** | **Наименование практического занятия** | **Описание** |
| **1** | Технология Java и среда разработки JetBrains IDEA | Технология Java. Введение | Загрузка лабораторного проекта, модифицирование кода по шаблону, запуск тестов, регистрация, архивация и отправка результата |
| **2** | Введение в объектно-ориентированное программирование на Java | Базовые элементы на ООП | Изучение базовых элементов ООП на языке Java |
| 3 | Типы данных, переменные, операторы | Синтаксис объявления переменных, использования литералов и операторов на языке Java | Изучение синтаксиса объявления переменных, использования литералов и операторов на языке Java |
| 4 | Создание и использование объектов. Классы и объекты String. Классы и объекты Java API | Синтаксис создания объектов и строк, сравнения, объединения строк, способы навигации по Java API | Изучение синтаксиса создания объектов и строк, сравнения, объединения строк, а также способов навигации по Java API |
| 5 | Операторы управления ходом программы | Операторы управления ходом программы | Изучение операторов управления ходом программы |
| 6 | Разработка методов, инкапсуляция и конструкторы, массивы объектов | Синтаксис определения методов и конструкторов, объявление массивов объектов, понятие инкапсуляции | Изучение синтаксиса определения методов и конструкторов, объявления массивов объектов, понятия инкапсуляции |
| 7 | Наследование, интерфейсы и полиморфизм | Синтаксис наследования, интерфейсов и полиморфизм | Изучение синтаксиса наследования, интерфейсов и полиморфизма |
| 8 | Механизм исключений и операции ввода/вывода | Механизм исключений и прикладные классы ввода-вывода | Изучение механизма исключений и прикладные классы ввода-вывода |
| 9 | Коллекции, стримы, лямбда | API коллекции и стримы, синтаксис лямбда | Изучение API коллекций и стримов, синтаксис лямбда |

**8.Оценочные материалы по образовательной программе**

**8.1. Вопросы тестирования по модулям**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ модуля** | **Вопросы входного тестирования** | **Вопросы промежуточного тестирования** | **Вопросы итогового тестирования** |
| **1**-9 | 1. Замените в классе Falcon слово "расширяет" на ключевое слово языка Java, такое чтобы класс Falcon наследовал классу Bird. Переопределите в классе Falcon унаследованный метод getName так, чтобы он возвращал строку "Falcon". Замените в классе Bird слово "реализует" на ключевое слово языка Java, такое чтобы класс Bird реализовывал интерфейс Flyer.  2. Метод sumOfNmbers класса Strings должен возвращать строковое представление суммы двух целочисленных аргументов, заданных в виде строк. Например, для аргументов add1="200" и add2="200", результат "400". Исправьте код метода sumOfNmbers, чтобы он работал правильно (изначально, он просто возвращает null). | ClassMap содержит карту рассадки учеников в классе в виде двумерного массива String[][] desks. Если ученик с именем Vasia сидит в 0-м ряду и 0-м столбце то значит desks[0][0]="Vasia" Реализуйте все методы в которых установлен комментарий TODO 1. Метод setDesk(String name) ищет в массиве первое свободное место со значением null и присваивает имя ученика. Возвращает строку с индексами в массиве. Например, "0\_0" если ученика разместили в desks[0][0]. Если мест нет то возвращает строку - "свободное место не найдено" 2. Метод setDesk(String name, int row, int col) размещает ученика на конкретное место, возвращает координату в виде "4\_5" (при row = 4, col = 5) если размещение успешно на соответствующее место. Если место занято, то - возвращает"место занято". 3. Метод searchDesk(String name) ищет где размещен ученик с заданным именем. Если ученик найден то возвращает координаты в виде "4\_5". В противном случае возвращает строку "ученик не найден". | Не проводится |
|  |  |  |  |
|  |  |  |  |

**8.2.**  **описание показателей и критериев оценивания, шкалы оценивания**

За выполнение каждой лабораторной работы слушателям выставляются баллы:

– лабораторные работы №№ 1-3: от 0 до 5 баллов;

– лабораторная работа № 4: от 0 до 15 баллов;

– лабораторные работы №№ 5-6: от 0 до 10 баллов;

– лабораторные работы №№ 7-10: от 0 до 25 баллов **.**

**8.3.**  **примеры контрольных заданий по модулям или всей образовательной программе**

Вопрос 1

Какие из нижеследующих утверждений об апплетах являются истинным?

Выберите один или несколько ответов:

a. апплет и приложение Java — это одно и то же. Любое приложение может быть апплетом, и наоборот.

b. апплет предназначен для распространения по сети Интернет и имеет ряд ограничений безопасности

c. апплет распространяется вместе с файлами HTML и отображается с помощью браузера

d. в отличие от приложения Java, апплет — это текстовый скрипт в составе HTML

e. и апплет и Java-приложение представляют собой байт-код

f. Java-апплет и JavaScript — это одно и тоже.

Вопрос 2

При запуске программы с помощью команды java HelloWorld.class было получено сообщение Exception in thread "main" java.lang.NoClassDefFoundError: Какова наиболее вероятная причина ошибки?

Выберите один ответ:

a. Не объявлен метод main

b. Класс HelloWorld объявлен в файле с другим именем

c. Имеется ошибка компиляции в классе HelloWorld

d. Не прописана переменная CLASS\_PATH

e. В команде запуска ошибочно указано расширение .class

f. HelloWorld.class отсутствует в данном каталоге, или в каталогах, указанных в CLASS\_PATH**.**

**8.4.**  **тесты и обучающие задачи (кейсы), иные практикоориентированные формы заданий**

1. Метод sumOfNmbers класса Strings должен возвращать строковое представление суммы двух целочисленных аргументов, заданных в виде строк.

Например, для аргументов "200" и "200", результат "400".

Максимальное значение аргумента "3000000000" (3 млрд.)

Исправьте код метода, чтобы он работал правильно.

Для проверки решения щелкните Check

вместо return null; должна быть правильно сформированная строка. Также и в других пунктах.

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

2. Метод initials класса Strings должен возвращать первые символы имени и фамилии в виде одной строки.

Например, для аргументов "Ivan" и "Boychenko", результат "IB".

Исправьте код метода, чтобы он работал правильно.

Для проверки решения щелкните Check

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

3. Метод equity класса Strings должен возвращать true если переданные строки равны без учета регистра.

Например, для аргументов "оранжевый" и "ОРАнЖевыЙ", результат true.

Исправьте код метода, чтобы он работал правильно.

Для проверки решения щелкните Check

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

4. Задание повышенной сложности. Напишите программу, которая по двум данным строкам определяет являются ли данные строки анаграммами. Анаграмма – это когда слово или фраза состоит полностью из тех же букв (и в том же количестве), что и другое слово или фраза. Задаваемые строки содержат только буквы и пробелы. Число пробелов в строках может быть разным. Пробелы должны игнорироваться.

Например, для аргументов "software" и "sw ear of t", результат true.

А, для аргументов "software" и "swearoftt", результат false, так как во втором аргументе есть лишняя t.

Исправьте код метода isAnagram класса AnagramChecker, чтобы он правильно определял являются ли аргументы анаграммами.

Для проверки решения щелкните Check

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

5. Метод dayOfWeek класса If должен принимать номер дня недели, а возвращать строковое название дня недели на русском.

Например, для аргумента 1, результат "Понедельник".

Например, для аргумента 2, результат "Вторник".

Например, для аргумента 7, результат "Воскресенье".

Если заданное число не попадает в диапазон 1..7 - возвращать "Ошибка данных"

Реализуйте код метода с помощью каскада проверок if-else-if.

Для проверки решения щелкните Check

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

6. Задание - аналог предыдущей задачи, чтобы показать взаимозаменяемость конструкций if и switch.

Метод dayOfWeek класса Switch должен принимать номер дня недели, а возвращать строковое название дня недели на русском.

Например, для аргумента 1, результат "Понедельник".

Для аргумента 2, результат "Вторник". ...

Для аргумента 7, результат "Воскресенье", и т.д.

Если заданное число не попадает в диапазон 1..7 - возвращать "Ошибка данных"

Реализуйте код метода с помощью оператора switch.

Для проверки решения щелкните Check

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

7. Метод numOfPlanet класса Switch должен принимать название планеты на русском, а возвращать ее порядковый номер от Солнца.

Учитываем только планеты с номером от 1 до 8 (до Нептуна). Плутон, согласно новой классификации, относится теперь к малым планетам, каковых в нашей звёздной системе тысячи, его (или её :)) не рассматриваем.

Например, для аргумента "Земля", результат 3.

Для аргумента "Юпитер", результат 5.

...

Для аргумента "Уран", результат 7, и т.д.

Если заданное имя неизвестно возвращать минус 1

Реализуйте код метода с помощью оператора switch.

Для проверки решения щелкните Check

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

8. Метод count класса While принимает длину возвращаемой последовательности в виде целого числа(int) и шаг последовательности также в виде целого числа(int), а возвращать строковое представление последовательности чисел, разделенное пробелами.

Последовательность – это арифметическая прогрессия с первым членом ряда 0 и далее идут числа меняющиеся с заданным шагом.

Например, для аргументов (5,2), результат "0 2 4 6 8".

Для аргументов (1, 3), результат "0" - то есть 0 это первый член ряда с длиной ряда 1.

...

Для аргументов (7, 3), результат "0 3 6 9 12 15 18".

Максимальная длина ряда 10, шаг от минус 5 до 7.

Реализуйте код метода с помощью оператора while или do-while (в зависимости как будет удобнее).

Конкатенацию строковых значений реализуйте с помощью StringBuilder (см. пример PersonTwo и видео раздела 4).

Алгоритм выглядит примерно так:

1. Создаем экземпляр StringBuilder sb = new StringBuilder();

2. Начинаем цикл

2.1. На каждом шаге вычисляем число ряда.

2.2. Добавляем полученное число в sb плюс пробел: sb.append("число" + " ");

3. Завершаем цикл.

4. Возвращаем результат с помощью sb.toString().trim(). trim нужен чтобы удалить лишние пробелы в начале и конце строки если такие появятся

Для проверки решения щелкните Check

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

9. Задание - аналог предыдущей задачи, чтобы показать взаимозаменяемость конструкций while и for. Метод count класса For принимает длину возвращаемой последовательности в виде целого числа(int) и шаг последовательности также в виде целого числа(int), а возвращать строковое представление последовательности чисел, разделенное пробелами.

Последовательность – это арифметическая прогрессия с первым членом ряда 0 и далее идут числа меняющиеся с заданным шагом.

Например, для аргументов (5,2), результат "0 2 4 6 8".

Для аргументов (1, 3), результат "0" - то есть 0 это первый член ряда с длиной ряда 1.

...

Для аргументов (7, 3), результат "0 3 6 9 12 15 18".

Максимальная длина ряда 10, шаг от минус 5 до 7.

Реализуйте код метода с помощью оператора for. Аналогично используйте StringBuilder для формирования выходной строки.

Для проверки решения щелкните Check

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

10. Заказ рубашек ShirtOrder при своем создании принимает число рубашек которое в него входит. Реализуйте метод getShirtQuantity() возвращающий число рубашек в заказе.

Реализуйте метод getShirt() который возвращает одну рубашку Shirt из заказа при этом уменьшая счетчик рубашек на единицу.

Пусть возвращаемый экземпляр Shirt создается в getShirt().

Если рубашек не осталось то getShirt() возвращает null.

Для проверки решения щелкните Check

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

11. Добавьте в класс Customer (Клиент) два варианта метода (перегрузка методов) с именем setInfo.

Добавленные методы должны выполнять следующее (в зависимости от списка параметров):

1. Устанавливать для клиента поля: ID – уникальный идентификатор, имя, почтовый адрес, и номер телефона (это минимальные сведения, необходимые для добавления клиента)

2. Устанавливать для клиента ID – уникальный идентификатор, имя, почтовый адрес, номер телефона и адрес e-mail

Реализуйте так чтобы один перегруженный метод вызывал другой.

Для проверки решения щелкните Check

setInfo(int id, String name, String address, String phone, String email)

setInfo(int id, String name, String address, String phone)

Ожидаемый результат: Тесты должны завершиться успешно с надписью Correct.

**8.5.**  **описание процедуры оценивания результатов обучения**

Итоговая аттестация проводится по накопительной схеме путем освоения каждой темы и выполнения лабораторно-практических работ.

Итоговая оценка рассчитывается на основании выполненных и зачтенных преподавателем результатов лабораторно-практических работ.

Слушатель, который по окончанию программы выполнит не менее 7 индивидуальных заданий и наберет не менее 95 баллов, считается прошедшим итоговую аттестацию.

Итоговая аттестация осуществляется преподавателем программы на основе двухбалльной («удовлетворительно» - «зачтено», «неудовлетворительно» - «не зачтено») системе оценок. **.**

**9.Организационно-педагогические условия реализации программы**

**9.1. Кадровое обеспечение программы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№**  **п/п** | **Фамилия, имя, отчество (при наличии)** | **Место основной работы и должность, ученая степень и ученое звание (при наличии)** | **Ссылки на веб-страницы с портфолио (при наличии)** | **Фото в формате jpeg** | **Отметка о полученном согласии на обработку персональных данных** |
| **1** | Бойченко Иван Валентинович | ОАО «ИнфоТеКС», разработчик, кандидат технических наук, доцент, сертифицированный инженер Novell (CNE) |  |  | Получено |

**9.2.Учебно-методическое обеспечение и информационное сопровождение**

|  |  |
| --- | --- |
| **Учебно-методические материалы** | |
| Методы, формы и технологии | Методические разработки,  материалы курса, учебная литература |
| Интерактивные лекции, лабораторные работы, консультации в режиме онлайн и оффлайн, технологии электронного и дистанционного обучения | 1. Электронный курс в системе Moodle ТУСУР: <https://udo.tusur.ru/course/view.php?id=53> |
|  |  |

|  |  |
| --- | --- |
| Информационное сопровождение | |
| Электронные  образовательные ресурсы | Электронные  информационные ресурсы |
| - Электронный учебник "Программирование на языке Java"  - Видеолекции по темам программы  - Базы данных | - Официальный веб-сайт Java: <http://www.oracle.com>  - Среда разработки на Java Eclipse: <http://www.eclipse.org/>  - Среда разработка на Java NetBeans: <https://netbeans.org/>  - Среда разработки на Java IntelliJ IDEA: <https://www.jetbrains.com/idea/>  - Java community process: [http://www.jcp.org](http://www.jcp.org/)  - Unified Modeling Language: <http://www.uml.org/> |
|  |  |

**9.3.Материально-технические условия реализации программы**

|  |  |
| --- | --- |
| Вид занятий | Наименование оборудования,  программного обеспечения |
| Лабораторные работы | Компьютер, подключенный к сети Интернет или ноутбук; интернет-браузер.  Перечень программного обеспечения: Microsoft Windows, OpenOffice, Kaspersky Endpoint Security 10 для Windows, 7-Zip, Google Chrome, Total Commander, JDK 1.8, JetBrains IDEA Edu, XMind 8 |
| Самостоятельная работа |
| Консультации в режиме онлайн | Компьютер, подключенный к сети Интернет или ноутбук; интернет-браузер, микрофон, наушники, веб-камера. |

**III.Паспорт компетенций (Приложение 2)**

ПАСПОРТ КОМПЕТЕНЦИИ

(Программирование на языке Java. Базовый курс)

(Федеральное государственное образовательное учреждение высшего образования "Томский государственный университет систем управления и радиоэлектроники")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Наименование компетенции | | Способность разрабатывать и отлаживать программный код | |
| 2. | Указание типа компетенции | общекультурная/  универсальная | - | |
| общепрофессиональная | - | |
| профессиональная | Да | |
| профессионально-специализированная | - | |
| 3. | Определение, содержание и основные сущностные характеристики компетенции | | ПК 1.1. Способность выполнять формализацию и алгоритмизацию поставленных задач  ПК 1.2. Способность написания программного кода с использованием языка программирования Java.  ПК 1.3. Способность оформления программного кода в соответствии с установленными требованиями.  ПК 1.4. Способность выполнять проверку и отладку программного кода. | |
| 4. | Дескриптор знаний, умений и навыков по уровням | | Уровни сформированности компетенции  обучающегося | Индикаторы |
|  | | Начальный уровень  (Компетенция недостаточно развита. Частично проявляет навыки, входящие в состав компетенции. Пытается, стремится проявлять нужные навыки, понимает их необходимость, но у него не всегда получается.) | - знает основные цели и области применения, основные понятия и компоненты технологии Java;  - знает базовые понятия языка Java;  - знает синтаксис языка Java;  - умеет устанавливать комплект разработки JDK и среду разработки Eclipse в операционной системе Windows или Linux;  - умеет компилировать и запускать программы на языке Java из командной строки |
|  | | Базовый уровень  (Уверенно владеет навыками, способен, проявлять соответствующие навыки в ситуациях с элементами неопределённости, сложности.) | - способен манипулировать ресурсами в среде разработки Eclipse;  - способен проектировать классы и применять механизмы наследования и полиморфизма;  - способен выражать отношения между классами с помощью нотации UML;  - умеет использовать механизм исключений при операциях ввода-вывода. |
|  | | Продвинутый  (Владеет сложными навыками, способен активно влиять на происходящее, проявлять соответствующие навыки в ситуациях повышенной сложности.) | - владеет навыками разработки и отлаживания кода Java в среде разработки Eclipse  - владеет навыком проектирования графического интерфейса пользователя с помощью библиотеки Swing;  - умеет решать задачи повышенной сложности. |
|  | | Профессиональный  (Владеет сложными навыками, создает новые решения для сложных проблем со многими взаимодействую-щими факторами, предлагает новые идеи и процессы, способен активно влиять на происходящее, проявлять соответствующие навыки  в ситуациях повышенной сложности.) | - |
| 5. | Характеристика взаимосвязи данной компетенции с другими компетенциями/ необходимость владения другими компетенциями для формирования данной компетенции | | Для освоения данной компетенции слушатели должны обладать общими компетенциями, включающими в себя способность:  ОК 1. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.  ОК 2. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.  ОК 3. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.  ОК 4. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации. | |
| 6. | Средства и технологии оценки | | Виртуальные лабораторные работы  Выполнение специальных заданий | |

**VI.Иная информация о качестве и востребованности образовательной программы** (результаты профессионально-общественной аккредитации образовательной программы, включение в системы рейтингования, призовые места по результатам проведения конкурсов образовательных программ и др.) (при наличии)

Отсутствует

**V.Рекомендаций к программе от работодателей**: наличие не менее двух писем и/или подтверждения на цифровой платформе Государственной системы предоставления ПЦС от работодателей о рекомендации образовательной программы для реализации в рамках Государственной системы предоставления ПЦС на формирование у трудоспособного населения компетенций цифровой экономики с указанием востребованности результатов освоения программы в сфере деятельности соответствующих компаний и готовности к рассмотрению заявок наиболее успешно освоивших образовательную программу граждан на прохождение стажировки и (или) собеседования на предмет трудоустройства путем проставления отметки в профиле программы

**VI.Указание на возможные сценарии профессиональной траектории граждан** по итогам освоения образовательной программы (в соответствии с приложением)

IT-индустрия, специалист в области Java-разработки**.** Java-программисты востребованы в тех областях, где используют проекты с многопользовательской обработкой данных и высокими требованиями к уровню безопасности.

**VII.Дополнительная информация**

**VIII.Приложенные Скан-копии**

Утвержденной рабочей программа (подпись, печать, в формате pdf)

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Томский государственный университет систем управления и радиоэлектроники» (ТУСУР)**

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Проректор по учебной работе  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_П.В. Сенченко  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_ г. |

**ПРОГРАММА**

**ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

**Программирование на языке Java. Базовый курс**

72 часа

СОГЛАСОВАНО**:**

|  |  |  |
| --- | --- | --- |
| Начальник УДО ИИ  Разработчик программы |  | А.В. Ковшов  И.В. Бойченко |

ТОМСК

2020

**I.ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ**

* 1. **Аннотация программы**

В современной IT-индустрии язык Java является одним из самых востребованных объектно-ориентированных языков программирования. Java стремительно развивается и охватывает практически все области программирования. Java используется для создания серверного программного обеспечения, прикладных программ, разработки веб-сайтов, игр и многого другого. Приложения, написанные на Java, могут работать на любом компьютере независимо от его архитектуры.

Java-разработчик должен обладать уверенными знаниями объектно-ориентированного программирования, уметь писать простой читабельный код с применением шаблонов проектирования. Java имеет множество библиотек и фреймворков. Разработчикам нужно знать их не только на уровне использования, но и понимать, как эти библиотеки и фреймворки устроены внутри.

Java-программисты востребованы в тех областях, где используют проекты с многопользовательской обработкой данных и высокими требованиями к уровню безопасности. Java-программист обычно разрабатывает архитектуру приложения, бизнес-логику приложения, поддерживает и сопровождает проект и пишет техническую документацию.

**1.2.** **Цель реализации программы**

Совершенствование и (или) получение новой компетенции, необходимой для профессиональной деятельности, и (или) повышение профессионального уровня в рамках имеющейся квалификации:

- способность применять основы языка программирования Java, необходимые для успешного освоения любой из высокоуровневых технологий семейства Java.

**1.3. Планируемые результаты обучения (РО)**

В результате освоения программы слушатель должен приобрести следующие знания, умения, опыт, необходимые для качественного изменения компетенций, указанных в п.1.2:

слушатель должен знать:

З1: историю возникновения и развития технологии Java, основные цели и области применения, основные понятия и компоненты технологии Java, состав комплекта разработки JDK, редакции и версии Java, жизненный цикл Java-программы;

З2: основные элементы интерфейса и возможности среды разработки Eclipse, возможности, появившиеся в Eclipse для интеграции новых возможностей Java;

З3: базовые понятия языка Java: класс, объект, атрибут, операция, композиция, наследование, интерфейс, пакет;

З4: синтаксис языка Java: идентификаторы, типы данных, массивы примитивных типов, переменные, арифметические и логические операторы и выражения;

З5: основные возможности класса String, классов Java API, массивы объектов.

слушатель должен уметь:

У1: устанавливать комплект разработки JDK и среду разработки Eclipse в операционной системе Windows или Linux;

У2: компилировать и запускать программы на языке Java из командной строки;

У3: манипулировать ресурсами в среде разработки Eclipse;

У4: проектировать классы и применять механизмы наследования и полиморфизма;

У5: выражать отношения между классами с помощью нотации UML;

У5: использовать механизм исключений при операциях ввода-вывода;

должен владеть (иметь практический опыт):

В1: разрабатывать и отлаживать код Java в среде разработки Eclipse;

В2: проектировать графический интерфейс пользователя с помощью библиотеки Swing.

**1.4. Программа разработана на основе:**

Профессионального стандарта 06.001 «Программист», Зарегистрированого в Министерстве юстиции Российской Федерации 18 декабря 2013 года, регистрационный N 30635

**1.5.** **Область применения программы (целевая аудитория)**

Программа предназначена для студентов и выпускников вузов и колледжей любых профилей, желающие получить новые знания и навыки, востребованные в IT-индустрии. Специалисты, желающие повысить свою квалификацию в области технологий разработки программного обеспечения. Все желающие начать карьеру Java-разработчика.

**1.6.** **Реализация программы**

Форма обучения: заочная

Применение ЭО и ДОТ: да

### 2. СОДЕРЖАНИЕ ПРОГРАММЫ

**2.1. Учебный план**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Наименование разделов/тем программы | Трудоемкость (час.) | Всего контактной работы  (час.) | в том числе (час.) | | | СРС (час.) | | Форма аттестации |
| ЛК | ПР | ЛБ | Изучение УММ | Выполнение заданий |
| 1. | Технология Java и среда разработки JetBrains IDEA | 7 | - | - | - | - | 3 | 4 |  |
| 2. | Введение в объектно-ориентированное программирование на Java | 5 | - | - | - | - | 2 | 3 |  |
| 3. | Типы данных, переменные, операторы | 5 | - | - | - | - | 2 | 3 |  |
| 4. | Создание и использование объектов. Классы и объекты String. Классы и объекты Java API | 5 | - | - | - | - | 2 | 3 |  |
| 5. | Операторы управления ходом программы | 7 | - | - | - | - | 2 | 5 |  |
| 6. | Разработка методов, инкапсуляция и конструкторы, массивы объектов | 13 | - | - | - | - | 4 | 9 |  |
| 7. | Наследование, интерфейсы и полиморфизм | 15 | - | - | - | - | 4 | 11 |  |
| 8. | Механизм исключений и операции ввода/вывода | 6 | - | - | - | - | 2 | 4 |  |
| 9. | Коллекции, стримы, лямбда | 9 | - | - | - | - | 2 | 7 |  |
|  | Итоговая аттестация | зачет | | | | | | | |
|  | **Итого (час.)** | **72** | - | - | - | - | **23** | **49** |  |

При необходимости программа может быть адаптирована под потребности заказчика.

## 2.2. Примерный календарный учебный график

Календарный учебный график формируется непосредственно при реализации программы и представлен в форме расписания занятий при наборе группы на обучение.

**2.3. Рабочая программа разделов**

|  |  |  |
| --- | --- | --- |
| Наименование компонентов программы | Содержание учебного материала и формы организации деятельности слушателей | Всего  (час.) |
|
| Тема 1. Технология Java и среда разработки JetBrains IDEA | *Видеолекции:*  1. Установка программного обеспечения  2. Запуск приложений из консоли  3. Запуск приложений из среды IDEA  4. Загружаем лабораторный проект  *Интерактивная лекция:*  Краткая история развития. Основные понятия Java. Java Software Development Kit (JDK). Редакции и нумерация версий Java. Эволюция Java. Жизненный цикл программы на языке Java. Компиляция и запуск приложений. Запуск Java-приложений с помощью интерпретатора java из командной строки. Приложения на Java. | 3 |
| *Лабораторная работа:*  Технология Java. Введение | 4 |
| Тема 2. Введение в объектно-ориентированное программирование на Java | *Видеолекция:*  Предварительная подготовка, определения ООП, примеры, выводы  *Интерактивная лекция:*  Что такое Объект? Что такое Класс? Что такое Наследование? Что такое Интерфейс? Что такое Пакет? | 2 |
| *Лабораторная работа:*  Базовые элементы на ООП | 3 |
| Тема 3.  Типы данных, переменные, операторы | *Видеолекция:*  Идентификаторы, типы данных, массивы примитивных типов, переменные, арифметические и логические операторы и выражения, литералы, бинарные литералы, символ подчеркивания в числовых литералах  *Интерактивная лекция:*  Идентификаторы. Типы данных и литералы. Бинарные литералы и использование символа подчеркивания в литералах. Приведение типов. Объявление и инициализация переменных. Соглашения о наименовании (Naming convention). Массивы примитивных типов. Операторы | 2 |
| *Лабораторная работа:*  Синтаксис объявления переменных, использования литералов и операторов на языке Java | 3 |
| Тема 4.  Создание и использование объектов. Классы и объекты String. Классы и объекты Java API | *Видеолекция:*  Создание и использование объектов. Классы и объекты String. Классы и объекты Java API  *Интерактивная лекция:*  Объявление, создание экземпляров и инициализация. Хранение ссылочных переменных и объектов в оперативной памяти. Использование класса String. Создание строковых переменных. Хранение строк в памяти. Спецификация классов Java API | 2 |
| *Лабораторная работа:*  Синтаксис создания объектов и строк, сравнения, объединения строк, способы навигации по Java API | 3 |
| Тема 5. Операторы управления ходом программы | *Видеолекция:*  Операторы управления ходом программы  *Интерактивная лекция:*  Оператор цикла while. Оператор цикла do-while. Оператор цикла for, и оператор запятая. Операторы break и continue. Имеющий дурную славу “goto”. Операторы условия if-else и возврата return. Оператор выбора switch | 2 |
| *Лабораторная работа:*  Операторы управления ходом программы | 5 |
| Тема 6. Разработка методов, инкапсуляция и конструкторы, массивы объектов | *Видеолекция:*  Разработка методов, инкапсуляция и конструкторы, массивы объектов  *Интерактивная лекция:*  Разработка методов в классах Java. Инкапсуляция и конструкторы. Массивы и объекты | 4 |
| *Лабораторная работа:*  Синтаксис определения методов и конструкторов, объявление массивов объектов, понятие инкапсуляции | 9 |
| Тема 7. Наследование, интерфейсы и полиморфизм | *Видеолекция:*  Наследование, интерфейсы и полиморфизм  *Интерактивная лекция:*  Наследование. Интерфейсы и абстрактные классы. Полиморфизм. Ключевое слово static. Ключевое слово final | 4 |
| *Лабораторная работа:*  Синтаксис наследования, интерфейсов и полиморфизм | 11 |
| Тема 8. Механизм исключений и операции ввода/вывода | *Видеолекция:*  Механизм исключений и операции ввода/вывода  *Интерактивная лекция:*  Механизм исключений. Файловый ввод\вывод. Сетевое взаимодействие. Нововведения Java 1.7: multi-catch, try-with-resources, выброс исключения с более точным указанием типа. | 2 |
| *Лабораторная работа:*  Механизм исключений и прикладные классы ввода-вывода | 4 |
| Тема 9. Коллекции, стримы, лямбда | *Видеолекция:*  Коллекции, стримы, лямбда  *Интерактивная лекция:*  Коллекции и генерики в Java: иерархия коллекций, оператор «алмаз» и выведение типов в генериках | 2 |
| *Лабораторная работа:*  API коллекции и стримы, синтаксис лямбда | 7 |
| Итоговая аттестация | Зачет по итогам выполнения лабораторных работ | - |

**2.4. Оценка качества освоения программы**

Итоговая аттестация проводится по накопительной схеме путем освоения каждой темы и выполнения лабораторно-практических работ.

Итоговая оценка рассчитывается на основании выполненных и зачтенных преподавателем результатов лабораторно-практических работ.

Слушатель, который по окончанию программы выполнит не менее 7 индивидуальных заданий, считается прошедшим итоговую аттестацию.

Итоговая аттестация осуществляется преподавателем программы на основе двухбалльной («удовлетворительно» - «зачтено», «неудовлетворительно» - «не зачтено») системе оценок.

Слушатель считается аттестованным, если имеет положительную оценку.

**2.4.2. Оценочные материалы**

Пример оценочного задания приведен в приложении А.

**2.4.3. Методические материалы**

1) «Положение об организации и осуществлении образовательной деятельности по программам дополнительного профессионального образования в ТУСУР» устанавливает регламентирующее формы, периодичность и порядок промежуточной аттестации слушателей.

2) «Положение об итоговой аттестации слушателей по программам дополнительного профессионального образования в ТУСУР» устанавливает: порядок организации итоговой аттестации в ТУСУР; порядок организации выполнения и защиты итоговых аттестационных работ, общие требования к итоговым аттестационным работам; общие критерии оценки знаний слушателей на итоговых аттестационных испытаниях.

**2.4.4. Анализ удовлетворенности требований слушателей**

До начала обучения по программе и по итогам обучения проводится анкетирование слушателей для анализа ожидания и удовлетворенности их требований к организации и качеству обучения.

**3. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ**

### 3.1. Материально-техническое обеспечение

|  |  |  |
| --- | --- | --- |
| Наименование  учебных помещений | Вид  занятий | Наименование оборудования,  программного обеспечения |
| Учебная аудитория для проведения занятий | Лекции, практические занятия, итоговая аттестация | - доска маркерная;  - мультимедийный проектор,  - комплект специализированной учебной мебели;  - рабочее место преподавателя и компьютеры слушателей:  - компьютер класса не ниже ПЭВМ INTEL Celeron D336 2.8ГГц. – до 20 шт.;  - компьютер подключен к сети «Интернет»;  - перечень программного обеспечения: Microsoft Windows, OpenOffice, Kaspersky Endpoint Security 10 для Windows, 7-Zip, Google Chrome. |

### 3.2. Учебно-методическое обеспечение и информационное обеспечение

Электронный учебный курс, разработанный в поддержку программы, размещенный в ЭИОС ТУСУР.

**3.3. Организация образовательного процесса**

Обучение организовано с применением образовательных технологий:

Дистанционные образовательные технологии, электронное обучение.

Консультации слушателям оказываются в форме онлайн – консультаций.

**3.4. Кадровое обеспечение образовательного процесса**

Кадровое обеспечение программы осуществляет преподавательский состав из числа преподавателей или сотрудников подразделений ТУСУР или привлеченных специалистов-практиков данной предметной области.

**3.5. Условия для функционирования электронной информационно-образовательной среды** (заполняется при реализации программы с использованием дистанционных образовательных технологий)

|  |  |  |
| --- | --- | --- |
| Электронные  информационные и образовательные ресурсы | Вид  занятий | Наименование оборудования,  программного обеспечения, необходимого слушателю для пользования ЭИОС |
| Система дистанционного обучения | СРС | Компьютер, подключенный к сети Интернет или ноутбук; интернет-браузер, микрофон, наушники, веб-камера.  Перечень программного обеспечения: Microsoft Windows, OpenOffice, Kaspersky Endpoint Security 10 для Windows, 7-Zip, Google Chrome. |

**4. РУКОВОДИТЕЛЬ И СОСТАВИТЕЛИ ПРОГРАММЫ**

**Руководитель программы:**

Ковшов Алексей Валерьевич, начальник управления дополнительного образования института инноватики ТУСУР

**Составитель:**

Бойченко Иван Валентинович

Кандидат технических наук, доцент, сертифицированный инженер Novell (CNE).

Область профессиональных интересов: распределенные вычисления, языки программирования.

В 1999–2012 гг. преподавал на факультете систем управления ТУСУР такие дисциплины, как «Программное обеспечение сетей ЭВМ», «Объектно-ориентированное программирование», «Сети ЭВМ». С 2016 г. работаю программистом в компании «Инфотекс» (специализация – распределенные приложения на Java).

Язык Java начал изучать в 2005 г. В 2006 г. перевел свои научные разработки на Java. Этой моей первой Java-разработкой пользуются уже более 10 лет в различных институтах РАН. В ходе обработки научных данных стало понятно, что будущее за распределенными вычислениями: кластерами, «облаками», big data и т. д. И язык Java вполне подходит на роль базовой технологии.

**Приложение А**

(справочное)

Вопрос 1

Какие из нижеследующих утверждений об апплетах являются истинным?

Выберите один или несколько ответов:

a. апплет и приложение Java — это одно и то же. Любое приложение может быть апплетом, и наоборот.

b. апплет предназначен для распространения по сети Интернет и имеет ряд ограничений безопасности

c. апплет распространяется вместе с файлами HTML и отображается с помощью браузера

d. в отличие от приложения Java, апплет — это текстовый скрипт в составе HTML

e. и апплет и Java-приложение представляют собой байт-код

f. Java-апплет и JavaScript — это одно и тоже.

Вопрос 2

При запуске программы с помощью команды java HelloWorld.class было получено сообщение Exception in thread "main" java.lang.NoClassDefFoundError: Какова наиболее вероятная причина ошибки?

Выберите один ответ:

a. Не объявлен метод main

b. Класс HelloWorld объявлен в файле с другим именем

c. Имеется ошибка компиляции в классе HelloWorld

d. Не прописана переменная CLASS\_PATH

e. В команде запуска ошибочно указано расширение .class

f. HelloWorld.class отсутствует в данном каталоге, или в каталогах, указанных в CLASS\_PATH