

Паспорт экзамена

по дисциплине «Технология разработки программного обеспечения», 2 семестр

1. Методика оценки

Экзамен проводится в письменной форме, по билетам. Билет состоит из 10 тестовых теоретических вопросов по темам дисциплины и одной задачи и формируется по следующему правилу:

- тестовые вопросы выбираются случайным образом из тестовой базы. Тестовые вопросы формируются в соответствии с теоретическими вопросами (темами), на каждый дается письменный ответ – 1-2 абзаца. Вес ответа 0.2 балла, сумма 0..20 баллов.
- задача – программа на Java или Scala в соответствии с формулировкой задачи – оценивается в 0...20 баллов.

Таким образом, проверяется уровень сформированности компетенций и соотношенных с ними индикаторов, закрепленных за дисциплиной.

На экзамене преподаватель вправе задавать студенту уточняющие и дополнительные вопросы из общего перечня (п. 4).

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к экзамену по дисциплине «Технология разработки программного обеспечения»

Тест 1. Тема: 16. Java. Рефлексия. Сериализация. JNI. Методы native.

Вопрос 3. Какой из форматов – XML или JSON является более эффективным при сериализации, почему?

Тест 2 Тема: 1. Производящие шаблоны builder, factory, prototype, singleton. Структурные шаблоны adapter, bridge, composite, decorator, facade, flyweight, proxy

Вопрос 3. Изобразите диаграмму классов или опишите ее (классы и отношения) для дерева в шаблоне composite. Как реализуется в этих классах метод рекурсивного обхода дерева?

Тест 3 Тема: 4. Java. Базовые типы данных. Класс String, его особенности и методы.

Классы – обертки для базовых типов данных (int и Integer), их назначение. Базовые типы

данных и ссылочные типы. Особенности работы со ссылочными типами, динамическое распределение памяти, «сбор мусора». Операции над ссылками.

Вопрос 9. Какие компоненты Java-программы создаются в динамической памяти, а какие являются статическими: класс, объект, статическое свойство класса, статический метод класса?

Тест 4 Тема: 16. Java. Рефлексия. Сериализация. JNI. Методы native.

Вопрос 6. Базовый класс DAO содержит средства на основе рефлексии, который позволяют добавлять, удалять, получать отдельные объекты и массивы объектов любого класса, произвольного от DAO. Опишите в общих словах работу класса DAO (4 балла).

Тест 5 Тема: 3. Шаблоны параллелизма Single Threaded Execution, Two-phase Termination, Asynchronous Task, Lock Object, Read/Write Lock, Scheduler, Double Buffering, Producer-consumer

Вопрос 6. Обоснуйте необходимость использования паттерна «поставщик-потребитель» (Producer-consumer) для воспроизведения видео из файла. Опишите логику его работы.

Тест 6 Тема: 10. Java. Поток и их программирование. Класс Thread и интерфейс Runnable. Связь потока с классом-прародителем. Прямое управление потоками (suspend, resume) и синхронизация потоков, ожидание событий (wait, notify).

Вопрос 6. Определите термины процесс и поток применительно к ОС. Каким образом процесс и поток реализуются в Java?

Тест 7 Тема: 12. Scala. Основные идеи. Архитектура. Синтаксис. Классы и объекты.

Прямое и обратное взаимодействие кода Scala – Java. Шаблоны (generic) как базовый элемент Scala. Массивы как объекты. Конструкторы. Синтаксис. Передача параметров при наследовании.

Вопрос 3. Какой способ определения типа – статический или динамический, используется в Scala? Как это согласуется со способом трансляции?

Тест 8 Тема: 7. Java. Наследование в Java. Древовидная система классов при одиночном наследовании. Класс Object, его назначение. Приведение типов ссылок при работе с системой наследуемых классов. Работа с данными базового класса. Доступ к базовому классу (super). Встроенный полиморфизм методов в Java. Интерфейс (interface). Использование интерфейсов для доступа к объектам различных классов. Наследование интерфейсов. Абстрактный базовый класс.

Вопрос 11. Определите термины Java «сужение» и «расширение». Определите механизм полиморфного вызова метода в этих терминах.

Тест 9 Тема: 14. Scala. Функции как параметры. Анонимные функции. Scala. Контейнеры типов (монады). Контейнеры Option для null-значения. Контроль наследования типов. Вариантность. Ковариантность и контрвариантность.

Вопрос 2. Определение и назначение Monad – шаблона контейнера для функционального преобразования

Тест 10 Тема: 1. Производящие шаблоны builder, factory, prototype, singleton. Структурные шаблоны adapter, bridge, composite, decorator, facade, flyweight, proxy

Вопрос 2. Имеется интерфейс FF с методом get() и класс AA, его присоединяющий.

Напишите заголовок класса BB, который замещает класс AA (конструктор, схему переопределения метода get)

Задача. Создать интерфейс с методом print. Создать 2 класса – один с целым числом, другой – с вещественным, оба с присоединенным интерфейсом. В main создать массив

ссылки на объекты обоих классов и в цикле вывести их содержимое через интерфейс методом print.

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись)
(дата)

2. Уровни освоения компетенций и критерии оценки

Ответ на экзаменационный билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный комплексный анализ материала, выявляет проблемы, предлагает механизмы их решения, представляет количественные характеристики определенных процессов, приводит конкретные примеры, не допускает ошибок и способен обосновать выбор метода решения задачи. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, сформированы в полном объеме. Оценка составляет *от 35 до 40 баллов*.

Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, дает характеристику процессов, явлений, проводит анализ причин, условий, способен представить количественные и качественные характеристики процессов, не допускает существенных ошибок при решении задачи. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, содержат несущественные пробелы и сформированы на базовом уровне. Оценка составляет *от 30 до 44 баллов*.

Ответ на экзаменационный билет засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, при решении задачи допускает непринципиальные ошибки, например, вычислительные. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, содержат пробелы и сформированы на пороговом уровне. Оценка составляет *от 20 до 29 баллов*.

Ответ на экзаменационный билет считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, при решении задачи допускает принципиальные ошибки. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, не сформированы. Оценка составляет *менее 20 баллов*.

3. Шкала оценки

Экзамен считается сданным, если сумма баллов по всем заданиям билета составляет от 20 до 40 баллов включительно. Сумма менее 20 баллов признается неудовлетворительным результатом промежуточной аттестации по дисциплине.

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, установленными в НГТУ.

4. Вопросы к экзамену по дисциплине «Технология разработки программного обеспечения»

Теоретические вопросы (темы)

1. Производящие шаблоны builder, factory, prototype, singleton. Структурные шаблоны adapter, bridge, composite, decorator, facade, flyweight, proxy
2. Поведенческие шаблоны command, iterator, mediator, snapshot, observer, state, strategy, method template, visitor. Системные шаблоны model-view-controller (MVC), session, transaction
3. Шаблоны параллелизма Single Threaded Execution, Two-phase Termination, Asynchronous Task, Lock Object, Read/Write Lock, Scheduler, Double Buffering, Producer-consumer
4. Java. Базовые типы данных. Класс String, его особенности и методы. Классы – обертки для базовых типов данных (**int** и Integer), их назначение. Базовые типы данных и ссылочные типы. Особенности работы со ссылочными типами, динамическое распределение памяти, «сбор мусора». Операции над ссылками.
5. Java. Наследование и приведение типов. Класс Object. Расширение и сужение ссылочного типа при переходе от класса к классу. Операция **instanceof**. Встроенный полиморфизм методов. Java. Объектная модель JVM. Классы Object и Class, их методы. Организация JVM. Система **мета-объектов** – описателей классов. Динамическая загрузка классов в JVM.
6. Java. Статические элементы классов. Их использование для задания констант и функций. Метод main. Модульная структура Java-программы. Пакет, класс, метод. Полные и сокращенные имена классов и методов. Объявление и подключение (импорт) пакетов. Области видимости: проект, пакет, класс, метод, производный класс. Модификаторы доступа (public, private, protected) и модификаторы изменения (static, final). Инициализаторы и конструкторы. Многообразие вариантов инициализации объектов. Явный вызов конструкторов. Массивы в Java. Массив как ссылочный тип и как объект. Реализация объекта-массива как динамического массив указателей (ссылок). Размерность массива, инициализация массивов. Многомерные массивы.
7. Java. Наследование в Java. Древоподобная система классов при одиночном наследовании. Класс Object, его назначение. Приведение типов ссылок при работе с системой наследуемых классов. Работа с данными базового класса. Доступ к базовому классу (super). Встроенный полиморфизм методов в Java. Интерфейс (interface). Использование интерфейсов для доступа к объектам различных классов. Наследование интерфейсов. Абстрактный базовый класс.
8. Java. Исключения, их генерация и обработка. Классы исключений. Секция finally.
9. Java. Программирование ввода-вывода. Классы потоков ввода-вывода. Классы - источники данных, классы представления данных. Форматы представления текста и особенности работы с ним в Java. Ввод-вывод текстовых файлов. Классы Reader, Writer. Буферизованный ввод-вывод. Понятие двоичного файла. Форматы внутреннего представления примитивных типов данных: целое со знаком и без, символ, вещественное. Двоичные потоки в Java. Объектовые потоки. **Сериализация**. Алгоритм сериализации системы связанных объектов.

10. Java. Потоки и их программирование. Класс Thread и интерфейс Runnable. Связь потока с классом-прародителем. Прямое управление потоками suspend, resume) и синхронизация потоков, ожидание событий (wait, notify).
11. Java. Вложенные и анонимные классы. Видимость текущего объекта и объекта-родителя. Использование вложенные классов для создания взаимодействующих потоков, CallBack-вызовов и обработчиков событий. Обработка событий в Java. Классы событий, интерфейсы слушателей, объявление слушателей событий. Способы создания обработчиков событий: использование общего класса-обработчика, классов-адаптеров, анонимных классов. Функциональное программирование. Лямбда-выражения в Java.
12. Scala. Основные идеи . Архитектура. Синтаксис. Классы и объекты. Прямое и обратное взаимодействие кода Scala – Java. Шаблоны (generic) как базовый элемент Scala. Массивы как объекты. Конструкторы. Синтаксис. Передача параметров при наследовании.
13. Scala. Внутренняя мета-модель языка. Функции как классы, константы как объекты, операции как методы (переопределение операций). Объектная модель языка. Классы Any, AnyVal, AnyRef. Приведение типов. Объектная модель языка. Trait-ы. Их отличие от интерфейсов.
14. Scala. Функции как параметры. Анонимные функции. Scala. Контейнеры типов (монады). Контейнеры Option для null-значения. Контроль наследования типов. Вариантность. Ковариантность и контрвариантность.
15. Scala. Структуры данных: Array, List, Map, кортеж. Операции над ними. Функциональные комбинаторы. Императивный и функциональный стиль программирования в Scala. Вложенные функции.
16. Java. Рефлексия. Сериализация. JNI
17. Сравнительная характеристика C++, Java, Scala
18. Разработка приложений для ОС Android на Java.

Задачи

1. «Перевернуть» содержимое строки (можно слить в массив символов, можно с использованием методов).
2. Базовый класс содержит целую переменную. Производный класс содержит вещественное, другой – строку. В main создается массив ссылок на 3 элемента разных классов. В цикле просматривается массив и при помощи instanceof проверяется принадлежность очередного элемента производному классу, производится сужение и вывод содержимого.
3. Создать 2 класса с произвольным содержимым. В третьем классе создать метод show, получающий объект Object и выполняющий любые известные действия над объектом (получение имени класса, вывод в текстовый поток – из классов Object и Class).
4. В одном пакете создать класс со статической переменной-константой и статическим методом. В другом пакете в main выполнить доступ к статическим элементам первого класса по короткому и по полному имени.
5. Класс двумерной точки инициализирует координату x нулем, а y – случайным числом от 0 до 10. Конструктор с двумя параметрами инициализирует обе координаты, конструктор с одним параметром инициализирует x из параметра, а y – нулем через

вызов предыдущего конструктора. Производный класс трехмерной точки использует конструктор двумерной точки, а координату z инициализирует сам.

6. Метод создает массив-треугольник заданной размерности (параметр метода), заполняет его нулями и возвращает в main, там он выводится.
7. Базовый класс содержит целое число (private) и второе целое – открытое. Метод inc увеличивает первое на 1. Производный класс содержит вещественное, его метод inc увеличивает все числа на 1 в обоих классах. Производный класс содержит строку, его метод inc добавляет к строке * и увеличивает целые в базовом классе. В main создать объекты всех классов и выполнить метод inc.
8. Создать интерфейс с методом print. Создать 2 класса – один с целым числом, другой – с вещественным, оба с присоединенным интерфейсом. В main создать массив ссылок на объекты обоих классов и в цикле вывести их содержимое через интерфейс методом print.
9. Объект содержит целое число. Метод add добавляет к нему целое – параметр. Если результат превышает 100, генерируется исключение. Класс-исключение получает значение текущего объекта до увеличения. Создать в main 2 объекта и продемонстрировать, что исключение срабатывает, и вывести в обработчике значение объекта, вызвавшего исключение.
10. Открыть файловый поток на заданном файле для ввода и байтовый поток для вывода. Скопировать файл в байтовый поток. Получить массив байтов. Создать на его основе байтовый поток ввода и открыть файловый поток для вывода. Скопировать. (Создание копии файла).
11. Прочитать текстовый файл посимвольно и построчно с постоянным именем и вывести в консоль.
12. Открыть двоичный файл и записать в него целое, вещественное, символ и строку. Показать его содержимое в 16-ричном дампе (far или totalCommander)
13. Создать 2 класса, в одном целое число, в другом – вещественное. Открыть объектовый поток и записать туда объект первого и объект второго классов. Открыть объектовый поток для чтения, прочитать эти объекты обратно и вывести содержимое.
14. В исходном классе есть переменная. Класс-поток при создании получает ссылку на исходный класс и в цикле через каждую секунду увеличивает эту переменную. Сделать 2 варианта класса – один с интерфейсом Runnable, другой – потомок от Thread
15. Класс-поток каждую секунду выводит в консоль сообщение. Основной оконный класс имеет 4 кнопки – запустить, приостановить, возобновить и уничтожить, через которые он управляет потоком напрямую.
16. Имеется два потока. Первый выводит строку, будит второго и засыпает. Второй выводит строку, будит первого и засыпает (вывод строк попеременно).
17. В основном классе имеется целая переменная. Вложенный класс-поток каждую секунду увеличивает ее значение на 1. Создать 2 объекта такого класса. Анонимный класс-поток уменьшает каждую секунду на 2 значение этой же переменной и выводит ее в консоль.
18. «Вручную» создать объект Button, разместить на экране, добавить его в оконный класс (все можно подсмотреть в генерируемом коде NetBeans для обычной кнопки) назначить ему обработчиком оконный класс и в нем прописать метод обработки, выводящий тест в консоль.

19. Аналогично, только класс-обработчик – вложенный.
20. Аналогично, только класс-обработчик – анонимный.
21. Создать между двумя приложениям (между двумя потоками в одном приложении) соединение через сокет и передать через него массив вещественных чисел.