Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники

Паспорт экзамена

по дисциплине «Конструирование и тестирование программного обеспечения», 7 семестр

1. Методика оценки

Экзамен проводится в устной форме по билетам. Билет формируется в виде 20 тестовых вопросов из разных тем (темы – из списка экзаменационные вопросов, тестовые вопросы выбираются случайным образом из закрытого списка. На каждый тестовый вопрос дается короткий письменный ответ 1-2 абзаца. Тест оценивается 0..2 балла, суммарно 0..40.

Таким образом, проверяются результаты обучения по дисциплине, соотнесенные с установленными в программе индикаторами достижения компетенций.

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 «Конструирование и тестирование программного обеспечения»

Тестовый вопрос 1 ----------
Тема: 8. 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.

6. Лямбда-выражение используется в Java как аналог анонимной функции. Объясните технологию ее определения с учетом того, что в Java отсутствует понятие функции как таковой, а есть метод класса.

Тестовый вопрос 2 ----------
Тема: 6. 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.

13. Опишите в виде последовательности действий или примером кода вывод двоичных данных с использование буферизации в ByteArrayOutputStream (4 балла).

Тестовый вопрос 3 ----------
Тема: 7. 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).

12. Что делает метод Thread.interrupt и как он используется?

Тестовый вопрос 4 ----------
Тема: 19. Понятие программной ошибки. Философия ошибок. Классификационные характеристики ошибок. Отладка, инспекция и Тестовый вопросирование как этапы поиска ошибок Ошибки вычислений и преобразований. Ошибки структурирования кода. Ошибки форматов входных данных. Ошибки форматов внутренних данных и соглашений по данным. Ошибки сборки, конфигурирования, размещения. Ошибки использования ресурсов. Ошибки, связанные с ограничениями по ресурсам. Ошибки реактивности и производительности. Ошибки параллелизма и синхронизации. Ошибки распределенных систем и протоколов. Ошибки пользовательского интерфейса.

4. Роль отладки, инспекции и тестирования

Тестовый вопрос 5 ----------
Тема: 9. Java. Рефлексия. Сериализация. JNI. Методы native.

7. Базовый класс DAO содержит средства на основе рефлексии, который позволяют записывать и читать из двоичного потока объекты любого класса, произвольного от DAO. Опишите в общих словах работу класса DAO, обработку полей примитивных типов, ссылок и массивов примитивных типов и ссылок (4 балла).

Тестовый вопрос 6 ----------
Тема: 3. 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.

4. Какие действия выполняет оператор import. Можно ли принципиально без него обойтись?

Тестовый вопрос 7 ----------
Тема: 1._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.

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

Тестовый вопрос 8 ----------
Тема: 21. Отладка и инспекция программного кода. Сущность отладки. Приемы отладки. Инспекция как неформальный анализ программного кода. Логический и временной анализ. Методы инспектирования. TDD – разработка, управляемая тестированием.

4. Определение пространства поиска дефекта на основе экспериментального тестирования программы

Тестовый вопрос 9 ----------
Тема: 7. 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. Опишите структуру и алгоритм работы паттерна «планировщик» в терминах – поток, засыпает, пробуждается, очередь, список, элемент запроса, исполняемый код, вставка/удаление из очереди (4 балла).

Тестовый вопрос 10 ----------
Тема: 12. Scala. Функции как параметры. Анонимные функции. Scala. Контейнеры типов (монады). Контейнеры Option для null-значения. Контроль наследования типов. Вариантность. Ковариантность и контрвариантность.

2. Определение и назначение Monade – шаблона контейнера для функционального преобразования
Тестовый вопрос 11 ----------
Тема: 6. 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.

8. Можно ли через FileInputStream открыть и напрямую читать тестовый файл в байтной кодировке? Если да, то какие при этом могут быть ограничения?

Тестовый вопрос 12 ----------
Тема: 1._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. Приведите пример операции расширения для базового типа данных. Является ли операция безопасной?

Тестовый вопрос 13 ----------
Тема: 10. 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.

6. Опишите сходства и различия шаблонов Scala с шаблонами Java и C++ с точки зрения особенностей компиляции.

Тестовый вопрос 14 ----------
Тема: 4. 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.

9. В базовом и производном классе определено поле (свойство) с одинаковым именем (int a;). Оба поля доступны в производном классе. Как их различить синтаксически?

Тестовый вопрос 15 ----------
Тема: 9. Java. Рефлексия. Сериализация. JNI. Методы native.

1. Определение рефлексии

Тестовый вопрос 16 ----------
Тема: 13. Scala. Структуры данных: Array, List, Map, кортеж. Операции над ними. Функциональные комбинаторы. Императивный и функциональный стиль программирования в Scala. Вложенные функции.

9. Почему для удаления изменения содержимого List необходимо присваивание?
```
var x = List[Int]()
x = 1 :: x
x = 3 :: x
x = x.appended(2)
```

Тестовый вопрос 17 ----------
Тема: 7. 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).

2. Имеется несколько потоков, выполняющих анимацию. Как при помощи wait/notify реализовать команды приостановки и возобновления анимации: алгоритм потока, данные, объект синхронизации, команды

Тестовый вопрос 18 ----------
Тема: 5. Java. Исключения, их генерация и обработка. Классы исключений. Секция finally.

1. Поясните термин «разматывание стека»

Тестовый вопрос 19 ----------
Тема: 12. Scala. Функции как параметры. Анонимные функции. Scala. Контейнеры типов (монады). Контейнеры Option для null-значения. Контроль наследования типов. Вариантность. Ковариантность и контрвариантность.

1. в точке вызова создается тело анонимной функции без заголовка. Какие способы определения формальных параметров существуют в Scala? Требуется ли задавать тип этих параметров?

Тестовый вопрос 20 ----------
Тема: 10. 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.

2. В каких случаях при определении переменной допустимо не указывать ее тип? Приведите примеры?

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

2. Критерии оценки результатов обучения, соотнесенных с уровнями освоения индикаторов достижения компетенций

Ответ на экзаменационный билет засчитывается на **продвинутом** уровне, если студент проводит сравнительный комплексный анализ материала, выявляет проблемы, предлагает механизмы их решения, представляет количественные характеристики определенных процессов, приводит конкретные примеры, не допускает ошибок. Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций не содержит пробелов. Установленные в программе компетенции 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. Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций содержит несущественные пробелы. Установленные в программе компетенции сформированы на базовом уровне. Оценка составляет *от 34 до 27 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. Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций содержит пробелы. Установленные в программе компетенции сформированы на пороговом уровне. Оценка составляет *от 26 до 20 баллов.*

Ответ на экзаменационный билет считается **неудовлетворительным** (ниже порогового уровня), если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений. Совокупность результатов обучения по дисциплине и соотнесенных с ними индикаторов достижения компетенций содержит существенные пробелы. Установленные в программе компетенции 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 «**Конструирование и тестирование программного обеспечения**»**

1. 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.
2. 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.
3. 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.
4. 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.
5. Java. Исключения, их генерация и обработка. Классы исключений. Секция finally.
6. 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.
7. 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).
8. 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.
9. Java. Рефлексия. Сериализация. JNI
10. 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.
11. 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.
12. Scala. Функции как параметры. Анонимные функции. Scala. Контейнеры типов (монады).

Контейнеры Option для null-значения. Контроль наследования типов. Вариантность. Ковариантность и контрвариантность.

13. Scala. Структуры данных: Array, List, Map, кортеж. Операции над ними. Функциональные комбинаторы. Императивный и функциональный стиль программирования в Scala. Вложенные функции.
14. Сравнительная характеристика C++,Java,Scala
15. Архитектура Android. Уровни. Их реализация, программирование, Java,C++. Реализация на основе Linux. Особенности виртуальной машины Dalvik. Файловая система: разделы внутренней памяти, логическая структура ФС. Особенности управления процессами и памятью. Особенности внутреннего программирования. Activity. События, жизненный цикл.
16. Android. Архитектура ОС Android. Ядро. Библиотеки. Среда исполнения (Dalvik-машина). Уровень каркаса приложений. Виды приложений (действия, сервисы, источники данных, приемники). Особенности компиляции приложений: файлы dx и apk. Система разработки для Android. Структура Android- приложения. Файл AndroidManifest.xml. Описание компонент (действий, служб) и разрешений
17. Android. Класс Activity. Классы элементов изображения и менеджеров компоновки View, Button, TextView, EditText, LinearLayout, TableLayout, TableRow. Статическое описание разметки в XML-файлах, доступ по идентификаторам ресурсов, назначение обработчиков событий. Динамическая (программная разметка). Диалоги с динамической (программной) и статической XML-разметкой
18. Android. Особенности выполнения приложений, связанные с ресурсными ограничениями. Класс Activity, его жизненный цикл, состояния и события. Запуск Activity, передача данных и возврат результата от Activity. Сохранение данных при выгрузке. Намерения, межпрограммные коммуникации. Широковещательные сообщения между Activity и приложениями.
19. Понятие программной ошибки. Философия ошибок. Классификационные характеристики ошибок. Отладка, инспекция и тестирование как этапы поиска ошибок Ошибки вычислений и преобразований. Ошибки структурирования кода. Ошибки форматов входных данных. Ошибки форматов внутренних данных и соглашений по данным. Ошибки сборки, конфигурирования, размещения. Ошибки использования ресурсов. Ошибки, связанные с ограничениями по ресурсам. Ошибки реактивности и производительности. Ошибки параллелизма и синхронизации. Ошибки распределенных систем и протоколов. Ошибки пользовательского интерфейса.
20. Структурное тестирование. Тестирование операторов, условий (решений) и путей. Комбинационное тестирование. Пример. Функциональное тестирование. Классы эквивалентности по входным данным. Проектирование тестового покрытия. Классы эквивалентности по граничным условиям. Примеры.
21. Отладка и инспекция программного кода. Сущность отладки. Приемы отладки. Инспекция как неформальный анализ программного кода. Логический и временной анализ. Методы инспектирования. TDD – разработка, управляемая тестированием.