Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Выборнова Любовь Анатольевна
Должность: Ректор
Дата подписания: 30.10.2026 08:54:23
Уникальный программный ключ:
c3b3b9c625f6c113afa2a2c42baff9e05a38b76e

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования «Поволжский государственный университет сервиса» (ФГБОУ ВО «ПВГУС»)

Колледж креативных индустрий и предпринимательства

УТВЕРЖДЕНО
на заседании ПЦК по образовательной программе 09.02.07 «Информационные системы и программирование»

Протокол от 24.10.2025 № 2

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

(фонд оценочных средств) для проведения промежуточной аттестации

по междисциплинарному курсу

учебному предмету, дисциплине, междисциплинарному курсу, профессиональному модулю

«РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»

наименование учебного предмета, дисциплины, междисциплинарного курса, профессионального модуля

по образовательной программе среднего профессионального образования –

программе подготовки специалистов среднего звена

программе подготовки специалистов среднего звена / программе подготовки квалифицированных рабочих, служащих
«Информационные системы и программирование»

наименование образовательной программы

специальность 09.02.07 «Информационные системы и программирование»

шифр, наименование специальности / профессии

Составитель Седнев Олег Геннадьевич, преподаватель Колледжа креативных индустрий и предпринимательства, к.с.н., доцент

Тольятти
2025

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ

1.1. Планируемые результаты обучения по междисциплинарному курсу

Код компетенции	Наименование компетенции
ПК 5.4	Производить разработку модулей информационной системы в соответствии с техническим заданием
ПК 5.5	Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы

В результате освоения междисциплинарного курса обучающийся должен:

иметь практический опыт в:

- управлении процессом разработки приложений с использованием инструментальных средств;

- обеспечении сбора данных для анализа использования и функционирования информационной системы;

- программировании в соответствии с требованиями технического задания;

- использовании критериев оценки качества и надежности функционирования информационной системы;

- применении методики тестирования разрабатываемых приложений;

- определении состава оборудования и программных средств разработки информационной системы;

- разработке документации по эксплуатации информационной системы;

- проведении оценки качества и экономической эффективности информационной системы в рамках своей компетенции;

- модификации отдельных модулей информационной системы.

уметь:

- осуществлять постановку задач по обработке информации; проводить анализ предметной области;

- осуществлять выбор модели и средства построения информационной системы и программных средств;

- использовать алгоритмы обработки информации для различных приложений;

- решать прикладные вопросы программирования и языка сценариев для создания программ;

- разрабатывать графический интерфейс приложения;

- создавать и управлять проектом по разработке приложения;

- проектировать и разрабатывать систему по заданным требованиям и спецификациям.

знать:

- основные виды и процедуры обработки информации, модели и методы решения задач обработки информации;

- основные платформы для создания, исполнения и управления информационной системой;

- основные процессы управления проектом разработки;

- основные модели построения информационных систем, их структуру, особенности и области применения;

- методы и средства проектирования, разработки и тестирования информационных систем;

- систему стандартизации, сертификации и систему обеспечения качества продукции.

1.2. Содержание междисциплинарного курса

№	Тема (раздел дисциплины) (в соответствии с РПД)	Код компетенции
		общие компетенции
1.	Тема 1. Основные инструменты для создания, исполнения и управления информационной системой.	ПК -5.4, ПК -5.5
2.	Тема 2. Разработка и модификация информационных систем.	

1.3. Система оценивания по междисциплинарному курсу

Курс изучается в течение двух семестров.

Форма промежуточной аттестации по междисциплинарному курсу 7 семестр – контрольная работа, 8 семестр – дифференцированный зачет.

Шкала оценки уровня сформированности компетенций		Шкала оценки уровня освоения междисциплинарного курса	
Уровневая шкала оценки компетенций	100 балльная шкала, %	100 балльная шкала, %	5-балльная шкала, дифференцированная оценка/балл
Допороговый	Ниже 61	Ниже 61	«Неудовлетворительно» / 2
Пороговый	61-85,9	61-69,9	«Удовлетворительно» / 3
		70-85,9	«Хорошо» / 4
Повышенный	86-100	86-100	«Отлично» / 5

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Задания для подготовки к промежуточной аттестации (контрольная работа)

7 семестр

Контрольная работа проводится для закрепления полученных знаний и направлена на расширение, углубление и систематизацию знаний, полученных при изучении МДК 05.02 «Разработка кода информационных систем» обучающимися по специальности 09.02.07 Информационные системы и программирование для промежуточной аттестации.

Содержание контрольной работы соответствует требованиям рабочей программы междисциплинарного курса и требованиям, изложенным в ФГОС СПО по специальности.

Цель контрольной работы состоит в установлении уровня подготовки студента к выполнению профессиональных задач и соответствия его подготовки требованиям ФГОС СПО по специальности 09.02.07 Информационные системы и программирование.

Контрольная работа состоит из ответов на вопросы открытого типа по темам междисциплинарного курса. Время на проведение контрольной работы – 40 минут.

ПК 5.4 - производить разработку модулей информационной системы в соответствии с техническим заданием

Задания для всех вариантов:

Задание 1.Разработать приложение на Java.

Задание 2. Проанализировать предметную область и выявить возможные функции для автоматизации их программной системой. Построить диаграмму вариантов использования для проектируемой системы.

ПК 5.5 - осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы

Задания для всех вариантов:

Задание 3. В соответствии с заданием на проектирование провести анализ предметной области, выявить основные классы и их взаимосвязи. Описать структуру проекта с помощью диаграммы классов.

Задание 4. Построить диаграммы последовательности для всех вариантов использования проектируемой системы

ПК 5.4 - производить разработку модулей информационной системы в соответствии с техническим заданием

Вариант 1. Разработка приложения для предметной области «Учѐт горюче-смазочных материалов на автозаправочной станции» с применением языка программирования Java.

Задание1. Разработать приложение на Java, позволяющее собирать и накапливать сведения о поступлении и реализации горюче-смазочных материалов на автозаправочных станциях г. Тольятти и Самарской области. Структура приложения обязательно должна включать следующие классы, созданные студентом: горюче-смазочный материал, автозаправочная станция и др.

Вариант 2. Разработка приложения для предметной области «Учѐт товаров в магазине» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения о поступлении и реализации товаров некоторого магазина. Структура приложения обязательно должна включать следующие классы, созданные студентом: товар, производитель, документ, поступление товара, реализация товара и др.

Вариант 3. Разработка приложения для предметной области «Организация учебного процесса в ВУЗе» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения об организации и диспетчеризации учебного процесса в ВУЗе. Структура приложения обязательно должна включать следующие классы, созданные студентом: академическая группа, специальность, дисциплина, аудитория, преподаватель и др.

Вариант 4. Разработка приложения для предметной области «Аквариум» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения о динамике растительного и животного мира, обитающего в аквариуме. Структура приложения обязательно должна включать следующие классы, созданные студентом: рыбы, породы рыб, растения (водоросли), разновидности растений, условия обитания (относятся к растениям и рыбам), питание и др.

Вариант 5. Разработка приложения для предметной области «Учѐт средств вычислительной техники на предприятии» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения по учѐту средств вычислительной техники. Структура приложения обязательно должна включать следующие классы, созданные студентом: средство вычислительной техники, компонент средства вычислительной техники (ВТ), производитель средства ВТ, продавец средства ВТ, гарантийное обслуживание, пользователь средства ВТ, структурное подразделение и др.

Вариант 6. Разработка приложения для предметной области «Учѐт основных средств предприятия» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения по учѐту основных средств предприятия. Структура приложения обязательно должна включать следующие классы, созданные студентом: карточка основного средства, группа основного средства, амортизация основного средства, списание основного средства, реновация основного средства, структурное подразделение и др.

Вариант 7. Разработка приложения для предметной области «Учѐт транспортных средств и их владельцев» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения по учѐту транспортных средств и их владельцев. Структура приложения обязательно должна включать следующие классы, созданные студентом: транспортное средство (ТС), категория транспортного средства, техосмотр, владелец ТС (юридическое или физическое лицо), топливо, модель ТС и др.

Вариант 8. Разработка приложения для предметной области «Учѐт рек с точки зрения народнохозяйственного значения» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения по учѐту рек, протекающих на территории некоторого региона. Структура приложения обязательно должна включать следующие классы, созданные студентом: реки, экологическая характеристика, народнохозяйственное значение, регионы, предприятия, другие водоѐмы и др.

Вариант 9. Разработка приложения для предметной области «Учѐт растений в ботаническом саду» с применением языка программирования Java.

Задание 1.Разработать приложение на Java, позволяющее собирать и накапливать сведения по учѐту растений, произрастающих на территории некоторого ботанического сада. Структура приложения обязательно должна включать следующие классы, созданные студентом: растение, семейство, вид, род, назначение, территория произрастания, карточка учѐта растений в ботаническом саду, совместимость с другими видами и др.

Вариант10. Разработка приложения для предметной области «Учѐт дикий животных в охотоведческом хозяйстве» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения по учѐту диких животных, обитающих на территории охотничьего хозяйства. Структура приложения обязательно должна включать следующие классы, созданные студентом: дикое животное, вид диких животных, класс животных, корм, лицензии на отстрел, карточка визуального учѐта животных и др.

Вариант11. Разработка приложения для предметной области «Учѐт домашних животных в фермерском хозяйстве» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения по учѐту домашних животных, содержащихся в фермерском хозяйстве. Структура приложения обязательно должна включать следующие классы, созданные студентом: домашнее животное, вид домашнего животного, порода, класс животных, корм, карточка учѐта животных, владелец и др.

Вариант 12. Разработка приложения для предметной области «Сведения о памятниках истории и архитектуры» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения о памятниках истории и архитектуры, представляющих историческую и

государственную ценность. Структура приложения обязательно должна включать следующие классы, созданные студентом: памятник, регион, создатель памятника, вид памятника, экспертная оценка состояния и др.

ПК 5.5 - осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы

Вариант13. Разработка приложения для предметной области «Сбор сведений о писателях и их литературных произведениях» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения о писателях и их литературных произведениях. Структура приложения обязательно должна включать следующие классы, созданные студентом: писатель (автор), произведение, жанр, страна (где проживал автор), язык (на котором написано или было переведено произведение), издание произведения (тираж), издательство и др.

Вариант 14. Разработка приложения для предметной области «Сведения о промышленных предприятиях» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения о промышленных предприятиях. Структура приложения обязательно должна включать следующие классы, созданные студентом: промышленное предприятие, форма собственности, сведения о владельце, продукция, партнѐры, конкуренты, отрасль экономики (к которой относится) и др.

Вариант 15. Разработка приложения для предметной области «Учѐт абитуриентов, поступающих в ВУЗ» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения об абитуриентах ВУЗа. Структура приложения обязательно должна включать следующие классы, созданные студентом: абитуриент, дисциплина, экзамен, специальность, экзаменационный лист, документ и др.

Вариант 16. Разработка приложения для предметной области «Математика» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и систематизировать сведения по разделам математики. Структура приложения обязательно должна включать следующие классы, созданные студентом: раздел математики (дисциплина), аксиомы, теоремы, леммы, сведения об учѐном-математике, задача, формула и др.

Вариант 17. Разработка приложения для предметной области «Учѐт вкладов, помещѐнных в банк» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения о деятельности отдела банка, работающего с вкладчиками. Структура приложения обязательно должна включать следующие классы, созданные студентом: вкладчик, вклад, тип вклада, сведения о банке, валюта, операции с вкладом и др.

Вариант 18. Разработка приложения для предметной области «Учѐт пациентов в регистратуре поликлиники» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения по учѐту пациентов в регистратуре городской поликлинике. Структура приложения обязательно должна включать следующие классы, созданные студентом: карточка больного (титульный лист), записи карточки, врач, журнал регистрации врача на дом, лекарство, заболевание и др.

Вариант 19. Разработка приложения для предметной области «Учѐт объектов библиотечного фонда» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения по учѐту книг в библиотеке. Структура приложения обязательно должна включать следующие классы, созданные студентом: карточка читателя (титульный лист), записи карточки, объект библиотечного фонда (книга, газета, журнал, …), издательство, карточка каталога, вид объекта библиотечного фонда, библиотечный раздел и др.

Вариант 20. Разработка приложения для предметной области «Учѐт заказов клиентов в ресторане» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения по учѐту заказов от клиентов в ресторане. Структура приложения обязательно должна включать следующие классы, созданные студентом: блюдо, группа блюд, заказ, клиент, столик, меню заказа (ценник) и др.

Вариант 21. Разработка приложения для предметной области «Учѐт клиентов и мобильных телефонов компании, предоставляющей услуги мобильной связи» с применением языка программирования Java.

Задание 1. Разработать приложение на Java, позволяющее собирать и накапливать сведения о клиентах и мобильных телефонах некоторой компании, предоставляющей услуги мобильной связи. Структура приложения обязательно должна включать следующие классы, созданные студентом: мобильный телефон, модель, владелец мобильного телефона, услуга, тариф, платѐж и др.

Вариант 22. Разработка приложения для предметной области «Деятельность страховой компании»с применением языка программирования Java.

Задание 1.Разработать приложение на Java, позволяющее собирать и накапливать сведения о финансовой деятельности компании. Учитывается, что страховая компания имеет множество филиалов по всей стране. При заключении договора на страхование необходимо зафиксировать дату заключения договора, страховую сумму, вид страхования, филиал в котором заключался данный договор, тарифную ставку. Также обратить внимание на то, что договоры заключают непосредственно страховые агенты, соответственно необходимо хранить всю информацию о агентах и филиале, в котором они работают.

Структура приложения обязательно должна включать следующие классы, созданные студентом: Договоры (Номер, дата заключения, страховая сумма, тарифная ставка, филиал, вид страхования), Вид страхования (Наименование), Филиал (Наименование, адрес, телефон)и др.

Вариант 23.Разработка приложения для предметной области «Деятельность ломбарда» с применением языка программирования Java.

Задание 1.Разработать приложение на Java, позволяющее собирать и накапливать сведения о финансовой деятельности компании. Задачей является отслеживание финансовой деятельности компании. К специалисту компании клиенты с целью получения денежных средств под залог определенных товаров. При обращении клиента специалист производит оценку стоимости и качества принесенного вам товара, определяет сумму, которую компания готова выдать клиенту и определяет свои комиссионные, также определяете срок возврата денежных средств. Если клиент согласен, то специалист спрашиваете данные клиента(ФИО, паспортные данные) и все договоренности фиксируются документально. В случае, если клиент в указанный срок не производит возврат денег, товар переходит в собственность ломбарда и может быть выставлен на продажу по цене меньшей или большей, чем была заявлена при сдаче.

Структура приложения обязательно должна включать следующие классы, созданные студентом:клиент (ФИО, номер, серия паспорта, дата выдачи паспорта, кемвыдан паспорт), категория товаров (Название, примечание), договор о сдаче в ломбард (Клиент, описание товара, дата сдачи, категория товара, дата возврата, сумма выданные средств, комиссионные)и др.

Вариант 24.Разработка приложения для предметной области «Деятельность банка по выдаче кредита» с применением языка программирования Java.

Задание 1.Разработать приложение на Java, позволяющее автоматизировать деяьельность руководителя информационно-аналитического отдела коммерческого банка. Обязанностью данного специалиста является отслеживание динамики кредитного отдела для юридических лиц. Кредит может получить юридическое лицо при предоставлении таких сведений: название, вид собственности, адрес, телефон, контактное лицо. Каждый кредит делится на несколько основных видов, в зависимости от процентной ставки, срока возврата и условий получения кредита, каждый из данных видов имеет свое название. Каждый факт выдачи кредита банком регистрируется, при этомфиксируется сумма кредита, клиент и дата выдачи.

Структура приложения обязательно должна включать следующие классы, созданные студентом: виды кредитов (Название, условия получения, ставка, срок), клиент (Название, вид собственности, адрес, телефон, контактное лицо), кредит (Вид кредита, клиент, сумма, дата выдачи).

Перечень вопросов и заданий для подготовки к (дифференцированному зачету)
8 семестр

ПК 5.4 - производить разработку модулей информационной системы в соответствии с техническим заданием

1. Структура CASE-средства.
2. Структура среды разработки. Основные возможности.
3. Основные инструменты среды для создания, исполнения и управления информационной системой.
4. Выбор средств обработки информации.
5. Организация работы в команде разработчиков.
6. Система контроля версий:совместимость, установка, настройка.

6. Обеспечение кроссплатформенности информационной системы.
7. Сервисно - ориентированные архитектуры.
8. Интегрированные среды разработки для создания независимых программ.
9. Особенности объектно-ориентированных и структурных языков программирования.
10. Разработка сценариев с помощью специализированных языков.
11. Обоснование и осуществление выбора модели построения или модификации информационной системы.
12. Обоснование и осуществление выбора средства построения информационной системы и программных средств.
13. Построение архитектуры проекта. Шаблон проекта.
14. Определение конфигурации информационной системы. Выбор технических средств.
15. Формирование репозитория проекта, определение уровня доступа в системе контроля версий. Распределение ролей.
16. Настройки среды разработки.

ПК 5.5 - осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы

17. Мониторинг разработки проекта. Сохранение версий проекта.
18. Требования к интерфейсу пользователя. Принципы создания графического пользовательского интерфейса (GUI).
19. Понятие спецификации языка программирования.
20. Синтаксис языка программирования.
21. Стиль программирования.
22. Основные конструкции выбранного языка программирования.
23. Описание переменных, организация ввода-вывода, реализация типовых алгоритмов.
24. Создание сетевого сервера и сетевого клиента.
25. Разработка графического интерфейса пользователя.
26. Отладка приложений.
27. Организация обработки исключений.
28. Виды, цели и уровни интеграции программных модулей.
29. Выбор источников и приемников данных, сопоставление объектов данных.
30. Транспортные протоколы. Стандарты форматирования сообщений.
31. Организация файлового ввода-вывода.
32. Процесс отладки. Отладочные классы.
33. Спецификация настроек типовой ИС.

Примерный тест для итогового тестирования:

ПК 5.4 - производить разработку модулей информационной системы в соответствии с техническим заданием

1. Выберите правильные утверждения:

-: класс — это тип данных;

-: объект класса может использоваться всюду, где используется объект подкласса;

-: объект класса можно создать только один раз;

-: на объект класса может не ссылаться объектная переменная.

2. Вставьте на место прочерка название одного из принципов ООП так, чтобы получилось верное определение:_____— это объединение данных и методов, предназначенных для
манипулирования этими данными в новом типе — классе.

-: наследование

-: полиморфизм

-: позднее связывание

-: инкапсуляция

3. Дан код:

```
String s; // 1
if ((s = "java") == "java") {// 2
        System.out.println (s+ " true");
} else {
        System.out.println (s+ " false");
}
```

Что будет результатом компиляции и запуска этого кода?

-: ошибка компиляции в строке 1, переменная не проинициализирована

-: ошибка компиляции в строке 2, неправильное выражение для оператора if

-: на консоль выведется java true

-: на консоль выведется java false

4. По типу результата все методы можно разделить на:
-:статические и не статические

-:константы и переменные

-:с возвращаемым значением и без

-:целочисленные и дробные

-:с аргументами и без

5. Какой класс является суперклассом для всех классов:
-:Main

-:MainClass

-:Object

-:Super

-:Class

6. В какой кодировке хранятся символы в переменной типа char:
-:UTF-32

-:UTF-16

-:UTF-8

-:ASCII

7. Результат сравнения - это:
-:Объект

-:Число

-:Символ

:Логическое значение

-:Строка

8. Что появится в консоли в результате работы программы:
<code>

byte a = 127;

a++;

System.out.println(a);

</code>

-:0

-:127

-:Ошибка времени исполнения

-:128

-:-128

9. Какой из способов объявления массива некорректен:
-:int a[][] = new int[5][5];

-:int[][] a = new int[5][5];

-: []int[] a = new int[5][5];

-:int[] a[] = new int[5][5];

10. Можно ли использовать оператор return в void методе:
-:Да

-:Нет

11. Являются ли два метода перегруженными, если у них одинаковые имена, разный набор аргументов и разный тип возвращаемого значения:
-:Нет

-:Да

12. Укажите правильную форму метода main:
-:public static void main(String[] args)

-:public void main(String[] args)

-:public static void main()

-:public static void Main(String[] args)

-:public int main(String[] args)

13. Какое значение будет сохранено в строке s?
<code>

String s = 2 + 2 + " = value";

</code>

-:ошибка компиляции

-:"4 = value"

-:"22 = value"

-:"2 + 2 = value"

-:"2 2 = value"

14. Как узнать количество элементов в массиве?
-:arrayName.size()

-:size(arrayName)

-:arrayName.length()

-:arrayName.length

-:length(arrayName)

15. Какойциклимеетинициализатор, условиеишаг?
-:for

-:infinite loop

-:do-while

-:for-earch

-:while

ПК 5.5 - осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы

16. Можнолипереопределять (override) privateметоды?
-:Нет

-:Да

17. Использование геттеров и сеттеров относится к принципу?
-:Инкапсуляция

-:Наследование

-:Полиморфизм

19. Инициализированная переменная в Java имеет:
-:Имя, первоначальное значение

-:Тип, имя

-:Имя

-:Тип, первоначальное значение

-:Тип, имя, первоначальное значение

20. Для объявления переменной в Java достаточно задать:
-:Тип, имя, первоначальное значение

-:Тип, имя

-:Тип, первоначальное значение

-:Имя

-:Имя, первоначальное значение

21. Для массивов в Java верно:
-:Изменяемый состав элементов разных типов

-:Фиксированное количество элементов разных типов

-:Фиксированное количество элементов одного типа

-:Изменяемый состав элементов одного типа

22. Может ли значение статического поля у объектов одного классы быть разным:
-:Да

+:Нет

23. Возможно ли унаследовать класс от двух суперклассов (множественное наследование):
-:Да

-:Нет

24. Какие категории типов данных поддерживаются в Java:
-:Переменные, примитивные

-:Ссылочные (объектные)

-:Примитивные

-:Примитивные, ссылочные (объектные)

-:Переменные, постоянные

25. Вызвать не статический метод или свойство из другого класса можно:

-:Только создав экземпляр класса, в котором они реализованы

-:Не создавая экземпляр класса, в котором они реализованы

-:Только из зависимых классов

-:Из любого класса в том же пакете

-:Из любого метода в этом же классе

26. К какому принципу ООП относятся понятия прав доступа и модификаторов public, protected, package-private и private?

-:Полиморфизм

-:Реализация

-:Абстракция

-:Наследование

-:Инкапсуляция

27. Можно ли в одном классе объявить больше одного метода main()?

-:Нет

-:Да

28. Возможно ли получить доступ к отдельным символам или диапазонам символов строки?

-:Да, символьные индексы начинаются с 'A'

-:Да, числовые индексы начинаются с 1

-:Да, символьные индексы начинаются с 'a'

-:Нет, доступ к отдельным символам строки получить нельзя

-:Да, числовые индексы начинаются с 0

29. Как JavaCodeConvention рекомендует называть классы и методы?

-:Классы пишутся заглавными буквами, методы целиком строчными

-:Классы пишутся строчными буквами, методы целиком заглавными

-:Классы начинаются со строчной буквы, методы с заглавной

-:Классы начинаются с заглавной буквы, методы тоже с заглавной

-:Классы начинаются с заглавной буквы, методы со строчной

30. Если в сигнатуре метода указано void, то ключевое слово return в теле метода:

-:Не обязательно, может быть указано с любым значением

-:Не обязательно, но может быть указано в качестве прерывания работы метода

-:Обязательно, но без какого-либо значения

-:Обязательно как минимум один раз с любым значением

-:Обязательно как минимум один раз без какого-либо значения