

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Рабочая программа дисциплины (модуля)
ПРОГРАММИРОВАНИЕ НА JAVA - ПРОДВИНУТЫЙ КУРС

Направление и направленность (профиль)
09.03.04 Программная инженерия. Программная инженерия

Год набора на ОПОП
2024

Форма обучения
очная

Владивосток 2025

Рабочая программа дисциплины (модуля) «Программирование на Java - продвинутый курс» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.04 Программная инженерия (утв. приказом Минобрнауки России от 19.09.2017г. №920) и Порядком организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утв. приказом Минобрнауки России от 06.04.2021 г. N245).

Составитель(и):

Гриняк В.М., доктор технических наук, профессор, Кафедра информационных технологий и систем, Viktor.Grinyak@vvsu.ru

Утверждена на заседании кафедры информационных технологий и систем от 30.05.2025 , протокол № 9

СОГЛАСОВАНО:

Заведующий кафедрой (разработчика)

Кийкова Е.В.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ	
Сертификат	1575633692
Номер транзакции	0000000000DC1211
Владелец	Кийкова Е.В.

1 Цель, планируемые результаты обучения по дисциплине (модулю)

Целью изучения дисциплины «Программирование на Java - продвинутый курс» является усвоение и закрепление основных приемов, методов и принципов работы при создании кроссплатформенных программ, усвоение навыков использования языка Java, подготовка к профессиональной сертификации.

Основные задачи изучения дисциплины:

- развитие навыков объектно-ориентированного программирования;
- освоение подходов к созданию консольных и визуальных кроссплатформенных программ;
- знакомство с понятиями и языком предметной области, в том числе международной англоязычной терминологией;
- развитие навыков работы в коллективе;
- подготовка к профессиональной сертификации Java-программиста.

Планируемыми результатами обучения по дисциплине (модулю), являются знания, умения, навыки. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы, представлен в таблице 1.

Таблица 1 – Компетенции, формируемые в результате изучения дисциплины (модуля)

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине	
			Код результата	Формулировка результата
09.03.04 «Программная инженерия» (Б-ИН)	ОПК-6 : Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.2к : Применяет современные программные среды программирования, разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ	РД1	Знание основных особенностей языка, принципов и приемов программирования, инструментальных средств программирования на Java, классификации платформ Java и возможностей их применения
			РД2	Умение создавать и тестировать программы на Java, в том числе для решения профессиональных задач с использованием возможностей инструментария среды программирования
			РД3	Навык использования приемов программирования, инструментария среды программирования Java и способов организации программ в том числе для решения прикладных задач

В процессе освоения дисциплины решаются задачи воспитания гармонично развитой, патриотичной и социально ответственной личности на основе традиционных российских духовно-нравственных и культурно-исторических ценностей, представленные в таблице 1.2.

Таблица 1.2 – Целевые ориентиры воспитания

Воспитательные задачи	Формирование ценностей	Целевые ориентиры
1 Формирование гражданской позиции и патриотизма		
2 Формирование духовно-нравственных ценностей		
3 Формирование научного мировоззрения и культуры мышления		
Развитие познавательного интереса и стремления к знаниям	Созидательный труд	Внимательность к деталям Способность находить, анализировать и структурировать информацию
4 Формирование коммуникативных навыков и культуры общения		

2 Место дисциплины (модуля) в структуре ОПОП

Дисциплина относится к обязательной части учебного плана.

Входными требованиями, необходимыми для освоения дисциплины, является наличие у обучающихся компетенций, сформированных при изучении дисциплин «Информатика и основы программирования», «Программирование на С», «Технология программирования».

3. Объем дисциплины (модуля)

Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающимися (по видам учебных занятий) и на самостоятельную работу, приведен в таблице 2.

Таблица 2 – Общая трудоемкость дисциплины

Название ОПОП ВО	Форма обуче- ния	Часть УП	Семестр (ОФО) или курс (ЗФО, ОЗФО)	Трудо- емкость	Объем контактной работы (час)						СРС	Форма аттес- тации
				(З.Е.)	Всего	Аудиторная			Внеауди- торная			
						лек.	прак.	лаб.	ПА	КСР		
09.03.04 Программная инженерия	ОФО	Б1.Б	4	3	55	18	36	0	1	0	53	ДЗ

4 Структура и содержание дисциплины (модуля)

4.1 Структура дисциплины (модуля) для ОФО

Тематический план, отражающий содержание дисциплины (перечень разделов и тем), структурированное по видам учебных занятий с указанием их объемов в соответствии с учебным планом, приведен в таблице 3.1

Таблица 3.1 – Разделы дисциплины (модуля), виды учебной деятельности и формы текущего контроля для ОФО

№	Название темы	Код результата обучения	Кол-во часов, отведенное на				Форма текущего контроля
			Лек	Практ	Лаб	СРС	
1	Синтаксис языка Java, классы в языке Java.	РД1	2	4	0	5	текущий опрос, отчет о выполнении практической работы
2	Наследование и инкапсуляция в языке Java.	РД1	1	2	0	4	текущий опрос, отчет о выполнении практической работы
3	Разработка классов в языке Java.	РД1	2	2	0	4	текущий опрос, отчет о выполнении практической работы
4	Наследование и интерфейсы в языке Java.	РД1	1	2	0	4	текущий опрос, отчет о выполнении практической работы
5	Обобщённые типы и коллекции значений в языке Java.	РД2	2	2	0	4	текущий опрос, отчет о выполнении практической работы
6	Работа со строками в языке Java.	РД2	2	4	0	4	текущий опрос, отчет о выполнении практической работы
7	Обработка исключений.	РД2	1	2	0	4	текущий опрос, отчет о выполнении практической работы
8	Ввод и вывод в Java программах. Файловый ввод и вывод.	РД2	1	2	0	4	текущий опрос, отчет о выполнении практической работы
9	Многопоточные программы Java.	РД3	1	4	0	4	текущий опрос, отчет о выполнении практической работы
10	Параллельное программирование Java.	РД3	1	4	0	6	текущий опрос, отчет о выполнении практической работы
11	Построение приложений баз данных с использованием JDBC API.	РД3	2	4	0	6	текущий опрос, отчет о выполнении практической работы
12	Локализация Java программ.	РД3	2	4	0	4	текущий опрос, отчет о выполнении практической работы
Итого по таблице			18	36	0	53	

4.2 Содержание разделов и тем дисциплины (модуля) для ОФО

Тема 1 Синтаксис языка Java, классы в языке Java. .

Содержание темы: Структура классов Java. Программные блоки и комментарии. Переменные. Условные операторы и циклы. .

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 2 Наследование и инкапсуляция в языке Java.

Содержание темы: Инкапсуляция при разработке классов Java. Моделирование задачи с использованием классов Java. Неизменяемые классы. Подклассы: создание и использование. Перегрузка методов класса. Методы с переменным числом аргументов.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 3 Разработка классов в языке Java.

Содержание темы: Спецификаторы доступа private, protected, default и public. Перегрузка конструкторов и других методов. Использование оператора instanceof для определения типа объекта. Виртуальный вызов методов класса. Преобразование типов «вверх» (апкостинг) и «вниз» (даункостинг). Перегрузка методов класса Object. Использование абстрактных классов. Ключевые слова final и static. Шаблон проектирования singleton. Вложенные классы.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 4 Наследование и интерфейсы в языке Java.

Содержание темы: Интерфейсы в Java, определение интерфейсов. Особенности использования интерфейсов и классов в программах. Расширение интерфейсов. Рефакторинг кода.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 5 Обобщённые типы и коллекции значений в языке Java.

Содержание темы: Обобщённые типы как способ создания классов в Java. Создание объектов в рамках обобщённого типа. Создание коллекций без использования обобщённых типов и с их использованием. Работа со структурами данных ArrayList, Set, HashMap. Реализация стека и очереди. Перечислимые типы.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 6 Работа со строками в языке Java.

Содержание темы: Чтение данных из командной строки. Поиск строк. Парсинг строк. Создание строк с использованием класса StringBuilder. Поиск в строке, парсинг строки и удаление строк с использованием регулярных выражений.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 7 Обработка исключений.

Содержание темы: Типы исключений в Java. Использование конструкций try и throw. Использование catch, единожды и многократно. Ключевое слово finally. Классы

исключений. Создание выборочных исключений и автозакрываемых ресурсов. Использование assertions.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 8 Ввод и вывод в Java программах. Файловый ввод и вывод.

Содержание темы: Основы ввода и вывода в Java программах. Чтение данных с консоли и вывод данных на консоль. Использование потоков для чтения и записи файлов. Чтение и запись объектов с использованием сериализации. Использование интерфейса Path для работы с файлами. Работа с классом Files для операций над файлами. Канальный и потоковый ввод-вывод в файлах. Работа с атрибутами файлов. Доступ к дереву каталогов. Поиск файлов с использованием класса PathMatcher.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 9 Многопоточные программы Java.

Содержание темы: Определение и создание потоков. Управление потоками. Синхронизация потоков. Проблемы многопоточного программирования.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 10 Параллельное программирование Java.

Содержание темы: Атомарные переменные. Метод ReentrantReadWriteLock(). Работа с коллекцией java.util.concurrent. Синхронизирующие классы. Использование ExecutorService. Fork-Join фреймворк.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 11 Построение приложений баз данных с использованием JDBC API.

Содержание темы: Основные функции JDBC API. Подключение к базе данных с использованием драйвера JDBC. Подача запросов получение результатов из базы данных. Транзакции и JDBC. Использование паттерна Data Access Object.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

Тема 12 Локализация Java программ.

Содержание темы: Особенности и задачи локализации программ. Определение и представление локализуемых данных. Чтение и установка локализуемых данных с помощью объекта Locale. Построение ресурсов. Вызов ресурсов из приложений. Форматирование текста и его локализация с использованием NumberFormat DateFormat.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим занятиям.

5 Методические указания для обучающихся по изучению и реализации дисциплины (модуля)

5.1 Методические рекомендации обучающимся по изучению дисциплины и по обеспечению самостоятельной работы

В ходе изучения дисциплины «Программирование на Java» студенты могут посещать аудиторные занятия (лекции, практические занятия, консультации).

Особое место в овладении частью тем данной дисциплины может отводиться самостоятельной работе, при этом во время аудиторных занятий могут быть рассмотрены и проработаны наиболее важные и трудные вопросы по той или иной теме дисциплины, а второстепенные и более легкие вопросы, а также вопросы, специфичные для ОПОП, могут быть изучены студентами самостоятельно.

В соответствии с учебным планом процесс изучения дисциплины может предусматривать проведение лекций, практических занятий, консультаций, а также самостоятельную работу студентов. Обязательным является проведение практических занятий в специализированных компьютерных аудиториях, оснащенных подключенными к центральному серверу терминалами или персональными компьютерами.

В рамках общего объема часов, отведенных для изучения дисциплины, предусматривается выполнение следующих видов самостоятельных работ студентов: подготовка к практическим занятиям, работа над рекомендованной литературой, самостоятельное решение задач.

Практикуется интерактивный способ работы над задачами: студенты размещают решение задачи на форуме, указанном преподавателем, проходит обсуждение задачи и ее решения, после чего преподаватель даёт свою оценку всем участникам процесса.

По тематике дисциплины выпущено большое количество печатных изданий. Электронные ресурсы могут быть использованы как источники справочной и обзорной информации по тематике дисциплины.

5.2 Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов

При необходимости обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) предоставляется учебная информация в доступных формах с учетом их индивидуальных психофизических особенностей:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания, консультации и др.

6 Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю)

В соответствии с требованиями ФГОС ВО для аттестации обучающихся на соответствие их персональных достижений планируемым результатам обучения по дисциплине (модулю) созданы фонды оценочных средств. Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 1.

7 Учебно-методическое и информационное обеспечение дисциплины (модуля)

0.1 Основная литература

0.2 Дополнительная литература

0.3 Ресурсы информационно-телекоммуникационной сети "Интернет", включая профессиональные базы данных и информационно-справочные системы (при необходимости):

Отсутствуют

8 Материально-техническое обеспечение дисциплины (модуля) и перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения

МИНОБРНАУКИ РОССИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Фонд оценочных средств
для проведения текущего контроля
и промежуточной аттестации по дисциплине (модулю)

ПРОГРАММИРОВАНИЕ НА JAVA - ПРОДВИНУТЫЙ КУРС

Направление и направленность (профиль)
09.03.04 Программная инженерия. Программная инженерия

Год набора на ОПОП
2024

Форма обучения
очная

Владивосток 2025

1 Перечень формируемых компетенций

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции
09.03.04 «Программная инженерия» (Б-ИН)	ОПК-6 : Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.2к : Применяет современные программные среды программирования, разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ

Компетенция считается сформированной на данном этапе в случае, если полученные результаты обучения по дисциплине оценены положительно (диапазон критериев оценивания результатов обучения «зачтено», «удовлетворительно», «хорошо», «отлично»). В случае отсутствия положительной оценки компетенция на данном этапе считается несформированной.

2 Показатели оценивания планируемых результатов обучения

Компетенция ОПК-6 «Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов»

Таблица 2.1 – Критерии оценки индикаторов достижения компетенции

Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине			Критерии оценивания результатов обучения
	Код	Тип	Результат	
ОПК-6.2к : Применяет современные программные среды программирования, разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ	РД 1	Знание	основных особенностей языка, принципов и приемов программирования, инструментальных средств программирования на Java, классификации платформ Java и возможностей их применения	достаточное для самостоятельного создания программ знание особенностей языка, приемов программирования, инструментальных средств программирования на Java
	РД 2	Умение	создавать и тестировать программы на Java, в том числе для решения профессиональных задач с использованием возможностей инструментария среды программирования	умение создавать и тестировать программы на Java
	РД 3	Навык	использования приемов программирования, инструментария среды программирования Java и способов организации программ в том числе для решения прикладных задач	навыки создания программ на Java, которые могут быть использованы для решения прикладных задач

Таблица заполняется в соответствии с разделом 1 Рабочей программы дисциплины (модуля).

3 Перечень оценочных средств

Таблица 3 – Перечень оценочных средств по дисциплине (модулю)

Контролируемые планируемые результаты обучения		Контролируемые темы дисциплины	Наименование оценочного средства и представление его в ФОС	
			Текущий контроль	Промежуточная аттестация
Очная форма обучения				
РД1	Знание : основных особенностей языка, принципов и приемов программирования, инструментальных средств программирования на Java, классификации платформ Java и возможностей их применения	1.1. Синтаксис языка Java, классы в языке Java.	Практическая работа	Тест
		1.2. Наследование и инкапсуляция в языке Java.	Практическая работа	Тест
		1.3. Разработка классов в языке Java.	Практическая работа	Тест
		1.4. Наследование и интерфейсы в языке Java.	Практическая работа	Тест
РД2	Умение : создавать и тестировать программы на Java, в том числе для решения профессиональных задач с использованием возможностей инструментария среды программирования	1.5. Обобщённые типы и коллекции значений в языке Java.	Практическая работа	Тест
		1.6. Работа со строками в языке Java.	Практическая работа	Тест
		1.7. Обработка исключений.	Практическая работа	Тест
		1.8. Ввод и вывод в Java программах. Файловый ввод и вывод.	Практическая работа	Тест
РД3	Навык : использования приемов программирования, инструментария среды программирования Java и способов организации программ в том числе для решения прикладных задач	1.9. Многопоточные программы Java.	Практическая работа	Тест
		1.10. Параллельное программирование Java.	Практическая работа	Тест
		1.11. Построение приложений баз данных с использованием JDBC API.	Практическая работа	Тест
		1.12. Локализация Java программ.	Практическая работа	Тест

4 Описание процедуры оценивания

Качество сформированности компетенций на данном этапе оценивается по результатам текущих и промежуточных аттестаций при помощи количественной оценки, выраженной в баллах. Максимальная сумма баллов по дисциплине (модулю) равна 100 баллам.

Вид учебной деятельности	Оценочное средство		
	Тестовые задания	Практические работы	Итого
Лекции	20		20
Практические занятия		80	80
Итого	20	80	100

Сумма баллов, набранных студентом по всем видам учебной деятельности в рамках дисциплины, переводится в оценку в соответствии с таблицей.

Сумма баллов по дисциплине	Оценка по промежуточной аттестации	Характеристика качества сформированности компетенции
от 91 до 100	«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций, обнаруживает всестороннее, систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями, умениями, применяет их в ситуациях повышенной сложности.
от 76 до 90	«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
от 61 до 75	«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по некоторым дисциплинарным компетенциям, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
от 41 до 60	«не зачтено» / «неудовлетворительно»	У студента не сформированы дисциплинарные компетенции, проявляется недостаточность знаний, умений, навыков.
от 0 до 40	«не зачтено» / «неудовлетворительно»	Дисциплинарные компетенции не сформированы. Проявляется полное или практически полное отсутствие знаний, умений, навыков.

5 Примерные оценочные средства

5.1 Примеры тестовых заданий

1. Given:

```
public class Sequence {
    Sequence() {
        System.out.print("x ");
    }
    public static void main(String[] args) {
        new Sequence().go();
    }
    void go() {
        System.out.print("y ");
    }
}
```

What is the result?

- A) x y
- B) y x
- C) x c
- D) y c

2. Given:

```
public class MyStuff {
    MyStuff(String n) {
        name = n;
    }
    String name;
    public static void main(String[] args) {
        MyStuff m1 = new MyStuff("guitar");
    }
}
```

```

        MyStuff m2 = new MyStuff("tv");
        System.out.println(m2.equals(m1));
    }
    public boolean equals(Object o) {
        MyStuff m = (MyStuff) o;
        if(m.name != null)
            return true;
        return false;
    }
}

```

What is the result?

- A) The output is "true" and MyStuff fulfills the Object.equals() contract.
- B) The output is "false" and MyStuff fulfills the Object.equals() contract.
- C) The output is "true" and MyStuff does NOT fulfill the Object.equals() contract.
- D) The output is "false" and MyStuff does NOT fulfill the Object.equals() contract
- E) Compilation fails

3. Given:

```

public class MyLoop {
    public static void main(String[] args) {
        String[] sa = {"tom ", "jerry "};
        for(int x = 0; x < 3; x++) {
            for(String s: sa) {
                System.out.print(x + " " + s);
                if( x == 1) break;
            }
        }
    }
}

```

What is the result?

- A) 0 tom 0 jerry 1 tom
- B) 0 tom 0 jerry 1 tom 1 jerry
- C) 0 tom 0 jerry 2 tom 2 jerry
- D) 0 tom 0 jerry 1 tom 2 tom 2 jerry
- E) 0 tom 0 jerry 1 tom 1 jerry 2 tom 2 jerry

4. Given:

```

class Feline {
    public String type = "f ";
    public Feline() {
        System.out.print("feline ");
    }
}

public class Cougar extends Feline {
    public Cougar() {
        System.out.print("cougar ");
    }
    public static void main(String[] args) {
        new Cougar().go();
    }
    void go() {
        type = "c ";
        System.out.print(this.type + super.type);
    }
}

```

```
}
```

What is the result?

- A) cougar c c
- B) cougar c f
- C) feline cougar c c
- D) feline cougar c f
- E) Compilation fails
- F) An exception is thrown at run time.

5. Given:

```
class Alpha {  
    String getType() {  
        return "alpha";  
    }  
}  
  
class Beta extends Alpha {  
    String getType() {  
        return "beta";  
    }  
}  
  
class Gamma extends Beta {  
    String getType() {  
        return "gamma";  
    }  
  
    public static void main(String[] args) {  
        Gamma g1 = new Alpha();  
        Gamma g2 = new Beta();  
        System.out.println(g1.getType() + " "  
            + g2.getType());  
    }  
}
```

What is the result?

- A) alpha beta
- B) beta beta
- C) gamma gamma
- D) alpha alpha
- E) Compilation fails.

6. Choose the False statement among the following?

Java support globally accessible variables

Java is programming language from Sun

JavaScript is a programming language from Netscape

None of the Above

7. The 16 bit coding scheme employed in Java programming is

EDCDIC

UNICODE

ASCII

Hexadecimal

8. What is the output of the following program?

```
public class example {  
    public static void main(String args[]) {  
        int x=0, y=2;  
        do {  
            x=++x;
```

```

        y--;
    } while(y>0);
    System.out.println(x);
}
}
0
1
2

```

Compilation Error

9. Command to execute a compiled java programs is

run

execute

javac

java

10. Choose the correct statement from the given choice?

Arrays in Java are essentially objects

It is not possible to assign one array to another. Individual elements of array can however be assigned.

Array elements are indexed from 1 to size of array

If a method tries to access an array element beyond its range, a compile warning is generated.

11. Consider the following code segment.

```
List list = new ArrayList();
```

```
list.add("P");
```

```
list.add("Q");
```

```
list.add("R");
```

```
list.set(2, "s");
```

```
list.add(2, "T");
```

```
list.add("u");
```

```
System.out.println(list);
```

What is printed as a result of executing the code segment?

(a) [P, Q, R, s, T]

(b) [P, Q, s, T, u]

(c) [P, Q, T, s, u]

(d) [P, T, Q, s, u]

(e) [P, T, s, R, u]

12. Consider the following instance variable and method.

```
private List nums;
```

```
/** Precondition: nums.size > 0
```

```
*/
```

```
public void numQuest() {
```

```
    int k = 0;
```

```
    Integer zero = new Integer(0);
```

```
    while (k < nums.size()) {
```

```
        if (nums.get(k).equals(zero))
```

```
            nums.remove(k);
```

```
        k++;
```

```
    }
```

```
}
```

Assume that List nums initially contains the following Integer values.

[0, 0, 4, 2, 5, 0, 3, 0]

What will List nums contain as a result of executing numQuest ?

(a) [0, 0, 4, 2, 5, 0, 3, 0]

(b) [4, 2, 5, 3]

(c) [0, 0, 0, 0, 4, 2, 5, 3]

(d) [3, 5, 2, 4, 0, 0, 0, 0]

(e) [0, 4, 2, 5, 3]

13. At a certain high school students receive letter grades based on the following scale.

Numeric Score Letter Grade

93 or above A

From 84 to 92 inclusive B

From 75 to 83 inclusive C

Below 75 F

Which of the following code segments will assign the correct string to grade for a given integer score ?

I.

```
if (score >= 93)
```

```
    grade = "A";
```

```
if (score >= 84 && score <= 92)
```

```
    grade = "B";
```

```
if (score >= 75 && score <= 83)
```

```
    grade = "C";
```

```
if (score < 75)
```

```
    grade = "F";
```

II.

```
if (score >= 93)
```

```
    grade = "A";
```

```
if (84 <= score <= 92)
```

```
    grade = "B";
```

```
if (75 <= score <= 83)
```

```
    grade = "C";
```

```
if (score < 75)
```

```
    grade = "F";
```

III.

```
if (score >= 93)
```

```
    grade = "A";
```

```
else if (score >= 84)
```

```
    grade = "B";
```

```
else if (score >= 75)
```

```
    grade = "C";
```

```
else
```

```
    grade = "F";
```

(a) II only

(b) III only

(c) I and II only

(d) I and III only

(e) I, II, and III

14. Consider the following declarations.

```
public interface Comparable {
```

```
    int compareTo(Object other);
```

```
}
```

```
public class SomeClass implements Comparable {
```

```
// ... other methods not shown
```

```
}
```

Which of the following method signatures of compareTo will satisfy the Comparable interface requirement?

I. public int compareTo(Object other)

II. public int compareTo(SomeClass other)

III. public boolean compareTo(Object other)

(a) I only

(b) II only

(c) III only

(d) I and II only

(e) I, II, and III

15. Consider the following instance variable and method.

```
private int[] myStuff;
```

```
/** Precondition: myStuff contains int values in no particular order.
```

```
*/
```

```
public int mystery(int num) {
```

```
    for (int k = myStuff.length - 1; k >= 0; k--) {
```

```
        if (myStuff[k] < num) {
```

```
            return k;
```

```
        }
```

```
    }
```

```
    return -1;
```

```
}
```

Which of the following best describes the contents of myStuff after the following statement has been executed?

```
int m = mystery(n);
```

(a) All values in positions 0 through m are less than n.

(b) All values in positions m+1 through myStuff.length-1 are less than n.

(c) All values in positions m+1 through myStuff.length-1 are greater than or equal to n.

(d) The smallest value is at position m.

(e) The largest value that is smaller than n is at position m.

16. Consider the following method.

```
/** Precondition: x >= 0
```

```
*/
```

```
public void mystery(int x) {
```

```
    System.out.print(x % 10);
```

```
    if ((x / 10) != 0) {
```

```
        mystery(x / 10);
```

```
    }
```

```
    System.out.print(x % 10);
```

```
}
```

Which of the following is printed as a result of the call mystery(1234) ?

(a) 1441

(b) 3443

(c) 12344321

(d) 43211234

(e) Many digits are printed due to infinite recursion.

17. Consider the following two classes.

```
public class Dog {
```

```
    public void act() {
```

```
        System.out.print("run");
```

```

        eat();
    }
    public void eat() {
        System.out.print("eat");
    }
}
public class UnderDog extends Dog {
    public void act() {
        super.act();
        System.out.print("sleep");
    }
    public void eat() {
        super.eat();
        System.out.print("bark");
    }
}

```

Assume that the following declaration appears in a client program.

```
Dog fido = new UnderDog();
```

What is printed as a result of the call `fido.act()` ?

- (a) run eat
- (b) run eat sleep
- (c) run eat sleep bark
- (d) run eat bark sleep
- (e) Nothing is printed due to infinite recursion.

18. Consider the following recursive method.

```

public static int mystery(int n) {
    if (n == 0)
        return 1;
    else
        return 3 * mystery(n - 1);
}

```

What value is returned as a result of the call `mystery(5)` ?

- (a) 0
- (b) 3
- (c) 81
- (d) 243
- (e) 6561

19. Consider the following instance variable and method.

```

private int[] arr;
/** Precondition: arr.length > 0
 */
public int checkArray() {
    int loc = arr.length / 2;
    for (int k = 0; k < arr.length; k++) {
        if (arr[k] > arr[loc])
            loc = k;
    }
    return loc;
}

```

Which of the following is the best postcondition for `checkArray` ?

- (a) Returns the index of the first element in array `arr` whose value is greater than `arr[loc]`
- (b) Returns the index of the last element in array `arr` whose value is greater than `arr[loc]`

- (c) Returns the largest value in array arr
 - (d) Returns the index of the largest value in array arr
 - (e) Returns the index of the largest value in the second half of array arr
20. Assume the following declarations have been made.

```
private String s;
private int n;
public void changer(String x, int y)
{
    x = x + "peace";
    y = y * 2;
}
```

Assume s has the value "world" and n is 6. What are the values of s and n after the call changer(s, n) ?

- (a) a is world, n is 6
- (b) a is worldpeace, n is 6
- (c) a is world, n is 12
- (d) a is worldpeace, n is 12
- (e) a is peace n is 12

Краткие методические указания

Промежуточный тест проводится в электронной форме во время последнего в учебном периоде практического занятия. Тест состоит из 20 тестовых заданий. На выполнение теста отводится 20 минут. Во время проведения теста использование литературы и других информационных ресурсов допускается только по предварительному согласованию с преподавателем.

Шкала оценки

Оценка	Баллы	Описание
5	19–20	Процент правильных ответов от 95% до 100%
4	16–18	Процент правильных ответов от 80 до 94%
3	13–15	Процент правильных ответов от 65 до 79%
2	9–12	Процент правильных ответов от 45 до 64%

5.2 Примеры заданий для выполнения практических работ

Вариант 1 Опишите класс с потоковой функцией, выводящей на экран ID потока, а затем с новой строки – текстовую строку, переданную в конструктор класса (она будет храниться как поле данных класса, строка должна быть достаточно большой, например, представлять собой отрывок из известного классического или современного текста). Напишите программу, которая создаёт несколько потоков (5 - 10). Каждый из созданных потоков выводит свой текст. При этом вывод текста синхронизирован (следующий текст не выводится, пока не закончится вывод предыдущего). Этого можно добиться, например, пометив метод, выводящий текст на консоль как synchronized.

Вариант 2 Написать программу со следующей функциональностью: Главный поток программы должен считывать вводимые пользователем строки и помещать их в конец списка ArrayList. Строки длиннее 80 символов можно разрезать на несколько строк. Если пользователь вводит пустую строку – программа останавливается и выводит на консоль текущее состояние списка. Дочерний поток пробуждается каждые пять секунд и сортирует получившийся список строк в лексикографическом порядке (используйте любую сортировку). Все операции над списком должны синхронизоваться при помощи synchronized.

Краткие методические указания

На выполнение одной практической работы отводится не более одного двухчасового занятия (не включая затраты времени на проведение промежуточного теста на последнем в учебном периоде практическом занятии). После выполнения каждой практической работы

студент должен представить отчет о ее выполнении, а также, по указаниям преподавателя, выполнить дополнительные практические задания по теме работы.

Шкала оценки

Оценка	Баллы	Описание
5	73–80	Студент демонстрирует умения на итоговом уровне: умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
4	61–72	Студент демонстрирует умения на среднем уровне: освоил основные умения, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.
3	49–60	Студент демонстрирует умения и навыки на базовом уровне: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных умений, навыков по дисциплинарной компетенции, испытываются значительные затруднения при оперировании умениями и при их переносе на новые ситуации.
2	33–48	Студент демонстрирует умения и навыки на уровне ниже базового: проявляется недостаточность умений и навыков.