

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Иркутский государственный университет путей сообщения»

Сибирский колледж транспорта и строительства

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПМ. 02 Проектирование управляющих программ компьютерных систем и
комплексов

МДК.02.03. Разработка прикладных приложений
для специальности

09.02.01 Компьютерные системы и комплексы

базовая подготовка
среднего профессионального образования

Иркутск 2023

Электронный документ выгружен из ЕИС ФГБОУ ВО ИрГУПС и соответствует оригиналу

Подписант ФГБОУ ВО ИрГУПС Трофимов Ю.А.

00a73c5b7b623a969ccad43a81ab346d50 с 08.12.2022 14:32 по 02.03.2024 14:32 GMT+03:00

Подпись соответствует файлу документа



Фонд оценочных средств разработан в соответствии с ФГОС СПО по специальности 09.02.01 Компьютерные системы и комплексы, утвержденного приказом Министерства Просвещения РФ от 25.05.2022 N 362, на основе рабочей программы ПМ.02 Проектирование управляющих программ компьютерных систем и комплексов МДК 02.03 Разработка прикладных приложений

РАССМОТРЕНО:

Цикловой методической
комиссией Компьютерные системы и
комплексы и информатика
Протокол №9
«26» мая 2023 г.
Председатель ЦМК: Арефьева Н.В.

СОГЛАСОВАНО:

Заместитель директора по УВР
А.П. Ресельс
«01» июня 2023 г.

Согласовано:

Ведущий специалист «Института
информационных технологий и
кибербезопасности »

 С.В. Бахвалов

Разработчик: Саквенко Т.В., преподаватель первой категории Сибирского колледжа транспорта и строительства ФГБОУ ВО «Иркутский государственный университет путей сообщения».

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	3
1 ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ	4
2.ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	122

1 ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Типы контроля успешности освоения ППССЗ обучающимися:

- входной контроль знаний;
- текущая аттестация;
- промежуточная аттестация;
- государственная итоговая аттестация.

Назначение фонда оценочных средств – оценить уровень подготовки студентов по МДК 02.03 Разработка прикладных приложений с целью установления их готовности к дальнейшему усвоению ППССЗ специальности 09.02.01 Компьютерные системы и комплексы.

Содержание фонда оценочных средств определяется в соответствии с ФГОС СПО специальности 09.02.01 Компьютерные системы и комплексы, рабочей программой МДК 02.03 Разработка прикладных приложений

В результате освоения МДК 02.03 Разработка прикладных приложений обучающийся должен:

Иметь практический опыт	<p>составления формализованных описаний решений поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов;</p> <p>разработки алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов;</p> <p>оценки и согласования сроков выполнения поставленных задач;</p> <p>создания программного кода в соответствии с техническим заданием (готовыми спецификациями);</p> <p>оптимизации программного кода с использованием специализированных программных средств;</p> <p>приведения наименований переменных, функций, классов, структур данных и файлов в соответствие с установленными в организации требованиями;</p> <p>структурирования и форматирования исходного программного кода в соответствии с установленными в организации требованиями;</p> <p>комментирования и разметки программного кода в соответствии</p>
------------------------------------	--

	<p>с установленными в организации требованиями;</p> <p>анализа и проверки исходного программного кода;</p> <p>отладки программного кода на уровне программных модулей;</p> <p>подготовки тестовых наборов данных в соответствии с выбранной методикой;</p> <p>регистрации изменений исходного текста программного кода в системе контроля версий;</p> <p>слияния, разделения и сравнения исходных текстов программного кода;</p> <p>сохранения сделанных изменений программного кода в соответствии с регламентом контроля версий;</p> <p>выполнения процедур сборки программных модулей и компонент в программный продукт;</p> <p>подключения программного продукта к компонентам внешней среды;</p> <p>проверки работоспособности выпусков программного продукта;</p> <p>внесения изменений в процедуры сборки модулей и компонент программного обеспечения, развертывания программного обеспечения, миграции и преобразования данных;</p> <p>разработки и документирования программных интерфейсов;</p> <p>разработки процедур сборки модулей и компонент программного обеспечения;</p> <p>разработки процедур развертывания и обновления программного обеспечения;</p> <p>разработки процедур миграции и преобразования (конвертации) данных;</p> <p>подготовки тестовых сценариев и тестовых наборов данных в соответствии с выбранной методикой;</p> <p>тестирования и верификации управляющих программ;</p> <p>оформления отчетов о тестировании;</p> <p>запуска процедуры установки прикладного программного обеспечения на конечных устройствах пользователей и/или серверном оборудовании;</p> <p>контроля процедуры установки прикладного программного обеспечения;</p> <p>настройки установленного прикладного программного обеспечения;</p> <p>обновления установленного прикладного программного</p>
--	--

	обеспечения.
Уметь	<p>использовать методы и приемы формализации задач;</p> <p>использовать методы и приемы алгоритмизации поставленных задач;</p> <p>использовать программные продукты для графического отображения алгоритмов;</p> <p>применять стандартные алгоритмы в соответствующих областях;</p> <p>применять выбранные языки программирования для написания программного кода;</p> <p>использовать выбранную среду программирования и средства системы управления базами данных;</p> <p>использовать возможности имеющейся технической и/или программной архитектуры;</p> <p>применять нормативные документы, определяющие требования к оформлению программного кода;</p> <p>применять инструментарий для создания и актуализации исходных текстов программ.</p> <p>выявлять ошибки в программном коде;</p> <p>применять методы и приемы отладки программного кода;</p> <p>интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов;</p> <p>применять современные компиляторы, отладчики и оптимизаторы программного кода;</p> <p>документировать произведенные действия, выявленные проблемы и способы их устранения;</p> <p>проводить оценку работоспособности программного продукта;</p> <p>создавать резервные копии программ и данных, выполнять восстановление, обеспечивать целостность программного продукта и данных;</p> <p>использовать выбранную систему контроля версий;</p> <p>выполнять действия, соответствующие установленному регламенту используемой системы контроля версий;</p> <p>интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов;</p> <p>применять современные компиляторы, отладчики и оптимизаторы программного кода;</p> <p>документировать произведенные действия, выявленные проблемы и способы их устранения;</p>

	<p>создавать резервные копии программ и данных, выполнять восстановление, обеспечивать целостность программного продукта и данных;</p> <p>выполнять процедуры сборки программных модулей и компонент в программный продукт;</p> <p>производить настройки параметров программного продукта и осуществлять запуск процедур сборки;</p> <p>писать программный код процедур интеграции программных модулей;</p> <p>использовать выбранную среду программирования для разработки процедур интеграции программных модулей;</p> <p>применять методы и средства сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов;</p> <p>разрабатывать и оформлять контрольные примеры для проверки работоспособности программного обеспечения;</p> <p>разрабатывать процедуры генерации тестовых наборов данных с заданными характеристиками;</p> <p>подготавливать наборы данных, используемых в процессе проверки работоспособности программного обеспечения;</p> <p>выявлять соответствие требований заказчиков к существующим продуктам;</p> <p>соблюдать процедуру установки прикладного программного обеспечения в соответствии с требованиями организации-производителя;</p> <p>идентифицировать инциденты, возникающие при установке программного обеспечения, и принимать решение по изменению процедуры установки.</p>
Знать	<p>методы и приемы формализации и алгоритмизации задач;</p> <p>языки формализации функциональных спецификаций;</p> <p>нотации и программные продукты для графического отображения алгоритмов;</p> <p>алгоритмы решения типовых задач, области и способы их применения;</p> <p>синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования;</p> <p>методологии разработки программного обеспечения;</p> <p>методологии и технологии проектирования и использования баз</p>

	<p>данных;</p> <p>технологии программирования;</p> <p>особенности выбранной среды программирования и системы управления базами данных;</p> <p>компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними;</p> <p>инструментарий для создания и актуализации исходных текстов программ;</p> <p>методы повышения читаемости программного кода;</p> <p>системы кодировки символов, форматы хранения исходных текстов программ;</p> <p>нормативные документы, определяющие требования к оформлению программного кода;</p> <p>методы и приемы отладки программного кода;</p> <p>типы и форматы сообщений об ошибках, предупреждений;</p> <p>способы использования технологических журналов, форматы и типы записей журналов;</p> <p>современные компиляторы, отладчики и оптимизаторы программного кода;</p> <p>сообщения о состоянии аппаратных средств;</p> <p>методы и средства верификации работоспособности выпусков программных продуктов;</p> <p>языки, утилиты и среды программирования, средства пакетного выполнения процедур;</p> <p>возможности используемой системы контроля версий и вспомогательных инструментальных программных средств;</p> <p>установленный регламент использования системы контроля версий;</p> <p>методы и средства сборки и интеграции программных модулей и компонент;</p> <p>интерфейсы взаимодействия с внешней средой;</p> <p>интерфейсы взаимодействия внутренних модулей системы;</p> <p>методы и средства сборки модулей и компонент программного обеспечения;</p> <p>интерфейсы взаимодействия с внешней средой;</p> <p>интерфейсы взаимодействия внутренних модулей системы;</p> <p>методы и средства разработки процедур для развертывания</p>
--	--

	<p>программного обеспечения;</p> <p>методы и средства миграции и преобразования данных;</p> <p>методы создания и документирования контрольных примеров и тестовых наборов данных;</p> <p>правила, алгоритмы и технологии создания тестовых наборов данных;</p> <p>требования к структуре и форматам хранения тестовых наборов данных;</p> <p>основные понятия в области качества программных продуктов;</p> <p>лицензионные требования по настройке устанавливаемого программного обеспечения;</p> <p> типовые причины инцидентов, возникающих при установке программного обеспечения;</p> <p>основы архитектуры, устройства и функционирования вычислительных систем;</p> <p>принципы организации, состав и схемы работы операционных систем;</p> <p>стандарты информационного взаимодействия систем.</p>
--	--

Содержание дисциплины должно быть ориентировано на овладение профессиональными компетенциями (ПК):

ПК 2.1. Проектировать, разрабатывать и отлаживать программный код модулей управляющих программ.

ПК 2.2. Владеть методами командной разработки программных продуктов.

ПК 2.3. Выполнять интеграцию модулей в управляющую программу.

ПК 2.4. Тестировать и верифицировать выпуски управляющих программ.

ПК 2.5. Выполнять установку и обновление версий управляющих программ

В процессе освоения дисциплины у студентов должны формировать общие компетенции (ОК):

ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;

ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 6 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 7 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.

Формой промежуточной аттестации по учебной дисциплине является Экзамен.

Формами текущей аттестации являются:

1. опрос
2. практическая работа
3. самостоятельная работа;
4. тестирование

Результаты освоения учебной дисциплины выражены в виде пятибалльной отметки.

2.ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

2.1 Назначение: ФОС предназначен для контроля и оценки промежуточных результатов освоения МДК.02.03. Разработка прикладных приложений

Формой промежуточной аттестации по учебной дисциплине является Экзамен.

Экзамен проводится в виде практической работы.

Количество вариантов для обучающихся 10.

Условия выполнения для обеспечения выполнения работы необходимо иметь компьютер со следующим программным обеспечением: операционная система Windows 10 и Java.

Время выполнения 60мин.

ТИПОВОЕ

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩЕГОСЯ

Вариант № __1__

Задание

Текст задания:

Формализовать поставленную задачу.

Написать программу решения задачи на языке Java

Отладить и протестировать программу.

Задача: Даны натуральные числа n, a_1, \dots, a_n . Определить количество членов a_k последовательности a_1, \dots, a_n , являющихся нечетными числами.

Инструкция по выполнению

Внимательно прочитайте задание.

Задание выполняется на Java, результаты работы программы проверяет экзаменатор.

Максимальное время выполнения задания – 60 мин.

ТИПОВОЕ

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩЕГОСЯ

Вариант № __2__

Задание

Текст задания:

Формализовать поставленную задачу.

Написать программу решения задачи на языке Java.

Отладить и протестировать программу.

Задача: Даны натуральные числа n, a_1, \dots, a_n . Определить количество членов a_k последовательности a_1, \dots, a_n , кратных 3 и не кратных 5.

Инструкция по выполнению

Внимательно прочитайте задание.

Задание выполняется на Java, результаты работы программы проверяет экзаменатор.

Максимальное время выполнения задания – 60 мин.

ТИПОВОЕ

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩЕГОСЯ

Вариант № __3__

Задание

Текст задания:

Формализовать поставленную задачу.

Написать программу решения задачи на языке Java. Отладить и протестировать программу.

Задача: Даны натуральные числа n, a_1, \dots, a_n . Определить количество членов a_k последовательности a_1, \dots, a_n , имеющих четные порядковые номера и являющихся нечетными числами.

Инструкция по выполнению

Внимательно прочитайте задание.

Задание выполняется на Java, результаты работы программы проверяет экзаменатор.

Максимальное время выполнения задания – 60 мин.

ТИПОВОЕ

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩЕГОСЯ

Вариант № __4__

Задание

Текст задания:

Формализовать поставленную задачу.

Написать программу решения задачи на языке Java.

Отладить и протестировать программу.

Задача: Даны целые числа a_1, \dots, a_n . Получить сумму тех чисел данной последовательности, которые кратны 5.

Инструкция по выполнению

Внимательно прочитайте задание.

Задание выполняется на Java, результаты работы программы проверяет экзаменатор.

Максимальное время выполнения задания – 60 мин.

ТИПОВОЕ

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩЕГОСЯ

Вариант № __5__

Задание

Текст задания:

Формализовать поставленную задачу – зарисовать блок схему алгоритма решения задачи.

Написать программу решения задачи на языке Java. Отладить и протестировать программу.

Задача: Даны целые числа a_1, \dots, a_{50} . Получить сумму тех чисел данной последовательности, которые нечетны и отрицательны.

Инструкция по выполнению

Внимательно прочитайте задание.

Задание выполняется на Java, результаты работы программы проверяет экзаменатор.

Максимальное время выполнения задания – 60 мин.

ТИПОВОЕ

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩЕГОСЯ

Вариант № __6__

Задание

Текст задания:

Формализовать поставленную задачу.

Написать программу решения задачи на языке Java. Отладить и протестировать программу.

Задача: Даны натуральное число n , действительные числа a_1, \dots, a_n . Получить удвоенную сумму всех положительных членов последовательности a_1, \dots, a_n .

Инструкция по выполнению

Внимательно прочитайте задание.

Задание выполняется на Java, результаты работы программы проверяет экзаменатор.

Максимальное время выполнения задания – 60 мин.

ТИПОВОЕ

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩЕГОСЯ

Вариант № __7__

Задание

Текст задания:

Формализовать поставленную задачу.

Написать программу решения задачи на языке Java.

Отладить и протестировать программу.

Задача: Даны натуральное число n , действительные числа a_1, \dots, a_n . В последовательности a_1, \dots, a_n все отрицательные члены увеличить на 0.5, а все неотрицательные заменить на 0.1.

Инструкция по выполнению

Внимательно прочитайте задание.

Задание выполняется на Java, результаты работы программы проверяет экзаменатор.

Максимальное время выполнения задания – 60 мин.

ТИПОВОЕ

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩЕГОСЯ

Вариант № __8__

Задание

Текст задания:

Формализовать поставленную задачу .

Написать программу решения задачи на языке Java. Отладить и протестировать программу.

Задача: Даны натуральное число n , целые числа a_1, \dots, a_n . Получить сумму положительных и отрицательных членов последовательности a_1, \dots, a_n .

Инструкция по выполнению

Внимательно прочитайте задание.

Задание выполняется на Java, результаты работы программы проверяет экзаменатор.

Максимальное время выполнения задания – 60 мин.

ТИПОВОЕ

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩЕГОСЯ

Вариант № __9__

Задание

Текст задания:

Формализовать поставленную задачу.

Написать программу решения задачи на языке Java. Отладить и протестировать программу.

Задача: Даны натуральное число n , действительные числа a_1, \dots, a_n . В последовательности a_1, \dots, a_n все члены, меньше двух, заменить нулями.

Инструкция по выполнению

Внимательно прочитайте задание.

Задание выполняется на Java, результаты работы программы проверяет экзаменатор.

Максимальное время выполнения задания – 60 мин.

ТИПОВОЕ

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩЕГОСЯ

Вариант № __10__

Задание

Текст задания:

Формализовать поставленную задачу .

Написать программу решения задачи на языке Java. Отладить и протестировать программу.

Задача: Даны натуральное число n , целые числа a_1, \dots, a_n . Найти наименьшее из чисел, входящих в последовательность $a_1 - 1, a_1, \dots, a_n$.

Инструкция по выполнению

Внимательно прочитайте задание.

Задание выполняется на Java, результаты работы программы проверяет экзаменатор.

Максимальное время выполнения задания – 60 мин.

ПАКЕТ ПРЕПОДАВАТЕЛЯ

Количество вариантов заданий для обучающихся: 10

Время выполнения каждого задания и максимальное время на промежуточную аттестацию:

Всего на зачет __60__ мин.

Условия выполнения заданий

Требования охраны труда: инструктаж по технике безопасности.

Оборудование: персональный компьютер со следующим обеспечением: операционная система Windows 10 и Java.

КРИТЕРИИ ОЦЕНКИ

Объектами оценки выступает продукт практической деятельности. Оценка и соответствующие критерии при этом основываются на эталонном качестве продукта.

Общими критериями оценки умений являются: правильность, точность, грамотность, качество, возможность применять навыки и умения на практике, наличие ошибок, их количество, характер и влияние на работу.

Критерии оценки задания

5 баллов ставится, если учащийся умеет грамотно:

- описывать исходные, выходные и промежуточные данные
- использовать команды языка Java.
- использовать синтаксис языка Java.
- включать в программу пояснения для пользователя программы
- тестировать и отлаживать программу

4 балла ставится, если в результате выполнения учащимся задания:

- алгоритм решения задачи является не оптимальным;
- отсутствуют пояснения для пользователя программы

3 балла ставится, если в результате выполнения учащимся задания:

- алгоритм решения задачи является не оптимальным;
- отсутствуют пояснения для пользователя программы
- пропущено описание исходных, выходных и промежуточных данных
- допущена 1 ошибка при написании программы

2 балла ставится, если учащийся:

- пропущено описание исходных, выходных и промежуточных данных
- допущены ошибки при написании программы
- программа не отлажена и не протестирована

2.2 Оценочные материалы для текущей аттестации

Назначение: ФОС предназначен для контроля и оценки текущих результатов освоения МДК.02.03. Разработка прикладных приложений

Контрольные тесты:

Тема 2. Введение в программирование на языке Java

Тест

1. Что такое метод без параметров в Java?
 - a) Метод, который не принимает на вход никаких параметров
 - b) Метод, который принимает неопределенное количество параметров
 - c) Метод, который возвращает значение

2. Как объявить метод с параметрами в Java?
 - a) `public void methodName()`
 - b) `public void methodName(int parameter)`
 - c) `public void methodName(parameter)`

3. Как передать параметры в метод в Java?
 - a) Они передаются после имени метода в круглых скобках
 - b) Они передаются в фигурных скобках в методе
 - c) Они передаются после ключевого слова `void`

4. В чем отличие метода без параметров от метода с параметрами?
 - a) Метод без параметров не принимает на вход никаких значений, а метод с параметрами – принимает
 - b) Метод без параметров всегда возвращает значение, а метод с параметрами нет
 - c) Метод без параметров может принимать неопределенное количество параметров, а метод с параметрами – только один

5. Что такое возвращаемое значение метода?

- a) Это значение, которое метод принимает на вход
 - b) Это значение, которое метод возвращает после выполнения
 - c) Это значение, которое метод передает другим методам
6. Как объявить метод, который возвращает целочисленное значение в Java?
- a) `public void methodName()`
 - b) `public int methodName()`
 - c) `public methodName(int value)`
7. Как вызвать метод с параметрами, если у нас есть значения, которые мы хотим передать в него?
- a) `methodName(5, "hello")`
 - b) `methodName`
 - c) `methodName(int value, String str)`
8. Что такое перегрузка методов?
- a) Возможность вызывать один и тот же метод с разным количеством параметров
 - b) Возможность вызывать методы из разных классов
 - c) Возможность создать метод с любым количеством параметров
9. Как объявить перегруженный метод в Java?
- a) Методы не могут быть перегружены в Java
 - b) Методы автоматически перегружены при объявлении с тем же именем, но разными параметрами
 - c) Методы могут быть перегружены только в том случае, если они возвращают разные типы данных
10. Какие методы не могут быть перегружены?
- a) Методы с разным количеством параметров

- b) Методы с разными типами возвращаемых значений
- c) Методы с одинаковыми именами

11. Что такое параметры метода?

- a) Это значения, которые метод возвращает
- b) Это значения, которые метод передает при вызове
- c) Это значения, которые метод сохраняет в переменных

12. Можно ли изменять значения переданных методу параметров внутри самого метода?

- a) Да, это возможно
- b) Нет, это приведет к ошибке компиляции
- c) Зависит от типа параметра и его модификатора

13. Какие модификаторы доступа могут быть у методов с параметрами?

- a) public, private, protected, default
- b) Нет модификаторов доступа у методов с параметрами
- c) Только public и private

14. Можно ли объявить метод, который не возвращает значение и не принимает параметры?

- a) Да, это возможно
- b) Нет, компилятор выдаст ошибку
- c) Такие методы называются конструкторами

15. Что такое аргументы метода?

- a) Это значения, которые метод возвращает
- b) Это значения, которые метод принимает при вызове
- c) Это значения, которые метод сохраняет в переменных

16. Какой оператор используется для вызова метода в Java?

- a) ->
- b) .
- c) ::

17. В чем заключается основное отличие между методом и функцией в Java?

- a) Функция в Java может принимать параметры, а метод – нет
- b) Метод связан с объектом, функция – нет
- c) Нет отличий, это синонимы

18. Как создать метод, который принимает переменное количество параметров?

- a) `public void methodName(int ... varArgs)`
- b) `public void methodName(varArgs[])`
- c) `public void methodName(int[] varArgs)`

19. Могут ли методы в Java возвращать несколько значений?

- a) Да, с помощью ключевого слова `return`
- b) Да, через запятую после ключевого слова `return`
- c) Нет, метод может вернуть только одно значение

20. Может ли метод быть вызван внутри другого метода?

- a) Да, это возможно
- b) Нет, методы могут быть вызваны только из мейн-метода
- c) Зависит от модификатора доступа у методов

Тема 3. Основные конструкции языка Java

Тест

1. Что такое оператор SWITCH в Java?
 - a) Оператор, используемый для работы с исключениями
 - b) Оператор, выполняющий различные действия в зависимости от значения выражения
 - c) Оператор, выполняющий циклический проход по набору элементов
 - d) Оператор, используемый для объявления переменных

2. Какой тип данных можно использовать в операторе SWITCH?
 - a) Только строковый
 - b) Только числовой
 - c) Любой, поддерживаемый примитивным типом данных
 - d) Только логический

3. Какая конструкция используется в операторе SWITCH для обработки значения по умолчанию?
 - a) default
 - b) case
 - c) break
 - d) continue

4. Что делает ключевое слово break в операторе SWITCH?
 - a) Завершает выполнение оператора SWITCH
 - b) Переходит к выполнению следующего case
 - c) Начинает выполнение оператора SWITCH
 - d) Отправляет значение по умолчанию

5. Какая структура данных часто используется в операторе SWITCH?

- a) Массивы
- b) Списки
- c) Хэш-таблицы
- d) Очереди

6. Какой синтаксис используется для цикла FOR в Java?

- a) `for i = 0; i < 5; i++`
- b) `for (int i = 0; i < 5; i++)`
- c) `for (i = 0; i < 5; i++)`
- d) `for (i = 0; i < 5)`

7. Какое ключевое слово используется для выполнения итерации в цикле FOR?

- a) `iterate`
- b) `loop`
- c) `continue`
- d) `break`

8. Что такое инициализация цикла FOR?

- a) Определение начального значения переменной цикла
- b) Определение условия окончания цикла
- c) Определение шага цикла
- d) Определение условия выполнения цикла

9. Как можно создать бесконечный цикл FOR?

- a) Указав бесконечное условие $i < \infty$
- b) Используя ключевое слово `infinite`
- c) Указав условие $i > 0$

d) Используя ключевое слово while

10. Какая конструкция используется для выполнения цикла WHILE в Java?

- a) while (condition) { }
- b) for (condition;) { }
- c) do { } while (condition)
- d) when (condition) { }

11. Что делает цикл WHILE?

- a) Выполняет итерацию до тех пор, пока условие истинно
- b) Выполняет итерацию определенное количество раз
- c) Выполняет итерацию только один раз
- d) Выполняет итерацию в обратном порядке

12. Как завершить выполнение цикла WHILE в середине итерации?

- a) Используя ключевое слово done
- b) Используя break
- c) Используя return
- d) Используя continue

13. Какой из циклов можно использовать, если необходимо выполнить итерацию хотя бы один раз?

- a) FOR
- b) WHILE
- c) DO-WHILE
- d) FOREACH

14. Какая конструкция используется для завершения текущей итерации и перехода к следующей в цикле WHILE?

- a) break

- b) continue
- c) return
- d) skip

15. Как изменить счетчик цикла в цикле FOR?

- a) Изменить исходное условие цикла
- b) Изменить значение переменной счетчика
- c) Использовать ключевое слово next
- d) Создать новый цикл

16. Какие практические применения цикла FOR в Java?

- a) Работа с коллекциями, массивами
- b) Обработка исключений
- c) Вывод на консоль
- d) Обработка строк

17. Что можно использовать в качестве условия цикла WHILE?

- a) Только логические выражения
- b) Только числовые значения
- c) Любые выражения, возвращающие значение типа boolean
- d) Строковые значения

18. Как завершить выполнение цикла DO-WHILE?

- a) Используя ключевое слово end
- b) Используя return
- c) Используя break
- d) Используя continue

19. Как изменить счетчик цикла WHILE?

- a) Нельзя изменить счетчик в цикле WHILE
- b) Изменить исходное условие цикла
- c) Изменить значение переменной счетчика
- d) Создать новый цикл

20. Какое ключевое слово используется для выполнения итерации в цикле DO-WHILE?

- a) do
- b) iterate
- c) continue
- d) next

Практическая работа № 4. Объявление и обработка одномерного массива.

Тест

1. Как объявить одномерный массив в Java?

- a) `int[] array;`
- b) `int array[];`
- c) `int array()[];`

2. Каков размер массива `array`?

`int[] array = new int[5];`

- a) 5
- b) 4
- c) 0

3. Как получить значение элемента массива с индексом 2?

- a) `array[1]`
- b) `array[2]`
- c) `array[3]`

4. Как изменить значение элемента массива с индексом 3 на 10?

- a) `array[3] = 10;`
- b) `array[2] = 10;`
- c) `array[4] = 10;`

5. Что произойдет, если попытаться обратиться к элементу массива, выходящему за пределы его размера?

- a) Будет выброшено исключение `ArrayIndexOutOfBoundsException`.
- b) Программа завершится с ошибкой.
- c) Будет изменено значение произвольного элемента массива.

6. Как вывести все элементы массива на экран?

- a) `for (int i = 0; i < array.length; i++) {System.out.println(array[i]);}`
- b) `for (int i = 1; i <= array.length; i++) {System.out.println(array[i]);}`
- c) `while (int i < array.length) {System.out.println(array[i]); i++;}`

7. Как посчитать сумму всех элементов массива?

- a) `int sum = 0; for (int i = 0; i < array.length; i++) {sum += array[i];}`
- b) `int sum = 0; for (int i = 1; i <= array.length; i++) {sum += array[i];}`
- c) `int sum = 0; while (int i < array.length) {sum += array[i]; i++;}`

8. Как отсортировать элементы массива по возрастанию?

- a) `Arrays.sort(array);`
- b) `Collections.sort(array);`
- c) `Arrays.sort(array, Collections.reverseOrder());`

9. Как найти максимальное значение в массиве?

- a) `int max = Arrays.max(array);`

- b) `int max = Collections.max(array);`
- c) `int max = Arrays.stream(array).max().getAsInt();`

10. Как найти индекс первого вхождения значения 5 в массиве?

- a) `int index = Arrays.indexOf(array, 5);`
- b) `int index = Arrays.asList(array).indexOf(5);`
- c) `int index = Arrays.stream(array).indexOf(5);`

11. Как скопировать элементы массива в новый массив?

- a) `int[] newArray = Arrays.copy(array);`
- b) `int[] newArray = array.clone();`
- c) `int[] newArray = Arrays.copyOf(array, array.length);`

12. Как проверить, содержит ли массив значение 7?

- a) `boolean contains = Arrays.contains(array, 7);`
- b) `boolean contains = Arrays.asList(array).contains(7);`
- c) `boolean contains = Arrays.stream(array).contains(7);`

13. Как удалить элемент с индексом 2 из массива?

- a) `array.remove(2);`
- b) `array.delete(2);`
- c) `array = ArrayUtils.remove(array, 2);`

14. Как найти среднее значение всех элементов массива?

- a) `double average = Arrays.average(array);`
- b) `double average = Arrays.stream(array).average().getAsDouble();`
- c) `double average = Arrays.mean(array);`

15. Как найти количество элементов массива, равных 4?

- a) `int count = Arrays.count(array, 4);`
- b) `int count = (int)Arrays.stream(array).filter(i -> i == 4).count();`
- c) `int count = Arrays.asList(array).count(4);`

16. Как добавить элемент со значением 6 в конец массива?

- a) `array.add(6);`
- b) `array.append(6);`
- c) `array = ArrayUtils.add(array, 6);`

17. Как создать массив из первых 5 чисел Фибоначчи?

- a) `int[] array = new int[] {1, 1, 2, 3, 5};`
- b) `int[] array = new int[5]; for (int i = 0; i < 5; i++) {array[i] = fibonacci(i);}`
- c) `int[] array = {1, 1, 2, 3, 5};`

18. Как найти сумму четных элементов массива?

- a) `int sum = 0; for (int i = 0; i < array.length; i++) {if (array[i] % 2 == 0) sum += array[i];}`
- b) `int sum = 0; for (int i = 1; i <= array.length; i++) {if (array[i] % 2 == 0) sum += array[i];}`
- c) `int sum = 0; while (int i < array.length) {if (array[i] % 2 == 0) sum += array[i]; i++;}`

19. Как найти минимальное значение в массиве?

- a) `int min = Arrays.min(array);`
- b) `int min = Collections.min(array);`
- c) `int min = Arrays.stream(array).min().getAsInt();`

20. Как найти индекс последнего вхождения значения 3 в массиве?

- a) `int index = Arrays.lastIndexOf(array, 3);`
- b) `int index = Arrays.asList(array).lastIndexOf(3);`

c) `int index = Arrays.stream(array).lastIndexOf(3);`

Практическая работа № 5. Объявление и обработка двумерного массива.

Тест

1. Что такое двумерный массив?

- a) Массив, содержащий только один элемент
- b) Массив, содержащий элементы разных типов данных
- c) Массив, содержащий элементы, которые сами являются массивами

2. Как объявить двумерный массив в Java?

- a) `int[] arr;`
- b) `int[][] arr;`
- c) `int arr[][];`

3. Как получить длину двумерного массива по горизонтали (количество строк)?

- a) `arr.length;`
- b) `arr[0].length;`
- c) `arr.length();`

4. Как получить длину двумерного массива по вертикали (количество столбцов)?

- a) `arr[0].length;`
- b) `arr.length;`
- c) `arr.columns();`

5. Как записать значение в элемент двумерного массива?

- a) `arr[0,0] = 5;`
- b) `arr[0][0] = 5;`
- c) `arr[0,0].value = 5;`

6. Как обратиться к элементу двумерного массива по индексам?

- a) `arr[index];`
- b) `arr[index1, index2];`
- c) `arr[index1][index2];`

7. Как вывести значения всех элементов двумерного массива?

- a) `for (int i = 0; i < arr.length; i++) { for (int j = 0; j < arr[i].length; j++) { System.out.println(arr[i][j]); } }`
- b) `System.out.println(arr);`
- c) `for (int i = 0; i < arr.length; i++) { System.out.println(arr[i]); }`

8. Как заполнить двумерный массив случайными числами?

- a) `for (int i = 0; i < arr.length; i++) { for (int j = 0; j < arr[i].length; j++) { arr[i][j] = Math.random(); } }`
- b) `Arrays.fill(arr, Math.random());`
- c) `for (int i = 0; i < arr.length; i++) { arr[i] = Math.random(); }`

9. Как проверить, является ли двумерный массив квадратным (с одинаковым количеством строк и столбцов)?

- a) `arr[0].length == arr.length;`
- b) `arr.length == arr.columns();`
- c) `arr.rows() == arr.columns();`

10. Как скопировать двумерный массив?

- a) `Arrays.copyOf(arr);`
- b) `Arrays.clone(arr);`
- c) `int[][] newArr = new int[arr.length][]; for (int i = 0; i < arr.length; i++) { newArr[i] = arr[i].clone(); }`

11. Как отсортировать элементы каждой строки двумерного массива в порядке возрастания?

- a) `Arrays.sort(arr);`
- b) `for (int i = 0; i < arr.length; i++) { Arrays.sort(arr[i]); }`
- c) `for (int i = 0; i < arr.length; i++) { Arrays.sort(arr); }`

12. Как найти наибольший элемент в двумерном массиве?

- a) `Arrays.max(arr);`
- b) `int max = arr[0][0]; for (int i = 0; i < arr.length; i++) { for (int j = 0; j < arr[i].length; j++) { if (arr[i][j] > max) max = arr[i][j]; } }`
- c) `int max = Arrays.stream(arr).max();`

13. Как найти сумму всех элементов двумерного массива?

- a) `Arrays.sum(arr);`
- b) `int sum = 0; for (int i = 0; i < arr.length; i++) { for (int j = 0; j < arr[i].length; j++) { sum += arr[i][j]; } }`
- c) `int sum = Arrays.stream(arr).sum();`

14. Как вывести каждый второй элемент каждой строки двумерного массива?

- a) `for (int i = 0; i < arr.length; i++) { for (int j = 0; j < arr[i].length; j+=2) { System.out.println(arr[i][j]); } }`
- b) `for (int i = 0; i < arr.length; i++) { System.out.println(arr[i][1]); }`
- c) `for (int i = 1; i < arr.length; i+=2) { System.out.println(arr[i]); }`

15. Как поменять местами строки двумерного массива?

- a) `Arrays.swapRows(arr, row1, row2);`
- b) `int[] temp = arr[row1]; arr[row1] = arr[row2]; arr[row2] = temp;`
- c) `arr[row1].swap(arr[row2]);`

16. Как поменять местами столбцы двумерного массива?

- a) `Arrays.swapColumns(arr, column1, column2);`

- b) `for (int i = 0; i < arr.length; i++) {int temp = arr[i][column1]; arr[i][column1] = arr[i][column2]; arr[i][column2] = temp;}`
- c) `arr[column1].swap(arr[column2]);`

17. Как проверить, содержит ли двумерный массив заданное значение?

- a) `Arrays.contains(arr, value);`
- b) `boolean contains = false; for (int i = 0; i < arr.length; i++) {for (int j = 0; j < arr[i].length; j++) {if (arr[i][j] == value) contains = true;}} return contains;`
- c) `boolean contains = Arrays.stream(arr).anyMatch(value);`

18. Как найти индексы первого вхождения заданного значения в двумерный массив?

- a) `Arrays.findFirstIndex(arr, value);`
- b) `for (int i = 0; i < arr.length; i++) {for (int j = 0; j < arr[i].length; j++) {if (arr[i][j] == value) return new int[]{i, j};}} return null;`
- c) `return Arrays.indexOf(arr, value);`

19. Как найти индексы всех вхождений заданного значения в двумерный массив?

- a) `Arrays.findAllIndexes(arr, value);`
- b) `List<int[]> indexes = new ArrayList<>(); for (int i = 0; i < arr.length; i++) {for (int j = 0; j < arr[i].length; j++) {if (arr[i][j] == value) indexes.add(new int[]{i, j});}} return indexes;`
- c) `return Arrays.indicesOf(arr, value);`

20. Как преобразовать двумерный массив в одномерный?

- a) `Arrays.flatten(arr);`
- b) `int[] newArr = new int[arr.length * arr[0].length]; int k = 0; for (int i = 0; i < arr.length; i++) {for (int j = 0; j < arr[i].length; j++) {newArr[k++] = arr[i][j];}} return newArr;`
- c) `return Arrays.stream(arr).flattened();`

Тема 4. Ввод данных из консоли

Ввод массивов.

Тест

1. Какой тип данных используется для объявления массива в Java?

- a) int
- b) double
- c) array
- d) char

2. Как объявить массив с 10 элементами в Java?

- a) `int[] arr = new int[10];`
- b) `arr[10] = new int[];`
- c) `arr = new int[10];`
- d) `int[10] arr;`

3. Как получить длину массива в Java?

- a) `arr.length();`
- b) `arr.len();`
- c) `arr.size();`
- d) `arr.length;`

4. Как присвоить значение элементу массива по индексу?

- a) `arr[1] = 10;`
- b) `set(arr, 1, 10);`
- c) `arr.set(1, 10);`
- d) `arr.set(10, 1);`

5. Как объявить и инициализировать массив одновременно?

- a) `int[] arr = new int[5];`
- b) `int[] arr = { 1, 2, 3, 4, 5};`
- c) `int arr[] = new int[]{ 1, 2, 3, 4, 5};`
- d) `int arr = { 1, 2, 3, 4, 5};`

6. Как получить значение элемента массива по индексу?

- a) `arr[1];`
- b) `arr.get(1);`
- c) `get(arr, 1);`
- d) `arr.get(1);`

7. Как скопировать значение одного массива в другой в Java?

- a) `Arrays.copy(arr1, arr2);`
- b) `arr1.copyTo(arr2);`
- c) `System.arraycopy(arr1, 0, arr2, 0, arr1.length);`
- d) `arr2 = arr1;`

8. Как отсортировать элементы массива в порядке возрастания в Java?

- a) `Arrays.sort(arr, DescendingOrder);`
- b) `arr.sort(AscendingOrder);`
- c) `Arrays.sort(arr);`
- d) `sort(arr);`

9. Как объединить два массива в Java?

- a) `Arrays.merge(arr1, arr2);`
- b) `System.arraycopy(arr1, 0, arr2, arr1.length, arr2.length);`
- c) `arr1.concat(arr2);`
- d) `Arrays.concat(arr1, arr2);`

10. Как обратить порядок элементов в массиве в Java?

- a) `arr.reverse()`;
- b) `Arrays.reverse(arr)`;
- c) `reverse(arr)`;
- d) `arr.flip()`;

11. Как найти максимальное значение в массиве в Java?

- a) `Math.max(arr)`;
- b) `arr.max()`;
- c) `Arrays.max(arr)`;
- d) `Arrays.stream(arr).max().getAsInt()`;

12. Как найти минимальное значение в массиве в Java?

- a) `Math.min(arr)`;
- b) `arr.min()`;
- c) `Arrays.min(arr)`;
- d) `Arrays.stream(arr).min().getAsInt()`;

13. Как проверить, содержит ли массив определенное значение в Java?

- a) `arr.contains(5)`;
- b) `Arrays.contains(arr, 5)`;
- c) `Arrays.stream(arr).contains(5)`;
- d) `arr.include(5)`;

14. Как удалить элемент из массива в Java?

- a) `remove(arr, 5)`;
- b) `arr.remove(5)`;
- c) `Arrays.delete(arr, 5)`;

d) `Arrays.copyOfRange(arr, 0, arr.length-1);`

15. Как проверить, пуст ли массив в Java?

a) `arr.isEmpty();`

b) `Arrays.isEmpty(arr);`

c) `arr.length() == 0;`

d) `arr == null;`

16. Как найти среднее значение элементов массива в Java?

a) `Arrays.average(arr);`

b) `arr.average();`

c) `Arrays.stream(arr).average().getAsDouble();`

d) `Math.average(arr);`

17. Как проверить, является ли массив отсортированным в Java?

a) `arr.isSorted();`

b) `Arrays.sorted(arr);`

c) `Arrays.isSorted(arr);`

d) `Arrays.stream(arr).isSorted();`

18. Как получить сумму элементов массива в Java?

a) `arr.sum();`

b) `Arrays.sum(arr);`

c) `Arrays.stream(arr).sum();`

d) `sum(arr);`

19. Как найти индекс первого вхождения элемента в массиве в Java?

a) `arr.findFirstIndex(5);`

b) `Arrays.findFirstIndex(arr, 5);`

- c) `Arrays.asList(arr).indexOf(5);`
- d) `Arrays.stream(arr).findFirstIndex(5);`

20. Как проверить, равны ли два массива в Java?

- a) `arr.equals(arr2);`
- b) `arr == arr2;`
- c) `Arrays.equals(arr, arr2);`
- d) `Arrays.deepEquals(arr, arr2);`

. Обработка строк: поиск, сравнение.

Тест

1. Каким методом можно определить длину строки в Java?

- a) `size()`
- b) `length()`
- c) `count()`
- d) `sizeOf()`

2. Каким методом можно проверить, содержит ли строка определенную подстроку?

- a) `contains()`
- b) `includes()`
- c) `find()`
- d) `containsSubstring()`

3. Что вернет метод `charAt(index)`, если `index` выходит за границы строки?

- a) Ошибка компиляции
- b) `Null`
- c) Пустую строку

d) Исключение `IndexOutOfBoundsException`

4. Какой метод используется для объединения двух строк в одну?

- a) `concat()`
- b) `merge()`
- c) `append()`
- d) `join()`

5. Какая из следующих операций вернет `true` при сравнении строк `"Java"` и `"java"`?

- a) `equals()`
- b) `equalsIgnoreCase()`
- c) `compareTo()`
- d) `compareToIgnoreCase()`

6. Какой метод используется для замены символов в строке?

- a) `replace()`
- b) `swap()`
- c) `modify()`
- d) `change()`

7. Что вернет метод `substring(startIndex, endIndex)` для строки `"Hello world"` при `startIndex = 6` и `endIndex = 11`?

- a) `"Hello"`
- b) `"world"`
- c) `"world"`
- d) `"d"`

8. Каким методом можно удалить пробельные символы в начале и конце строки?

- a) trim()
- b) deleteWhitespace()
- c) removeSpaces()
- d) clean()

9. Какой метод используется для преобразования строки в символьный массив?

- a) toCharArray()
- b) toStringArray()
- c) toArray()
- d) castToArray()

10. Что вернет метод indexOf(substring) для строки "Hello world" при substring = "world"?

- a) -1
- b) 0
- c) 6
- d) 11

11. Каким методом можно разделить строку на массив строк по определенному разделителю?

- a) split()
- b) divide()
- c) separate()
- d) breakApart()

12. Какой метод используется для преобразования строки в верхний регистр?

- a) toUpperCase()
- b) capitalize()
- c) upperCase()

d) `convertToUpperCase()`

13. Что вернет метод `startsWith(prefix)` для строки "Hello world" при `prefix = "Hello"`?

a) `true`

b) `false`

c) `0`

d) "Hello world"

14. Каким методом можно проверить, пустая ли строка?

a) `isEmpty()`

b) `isNull()`

c) `isBlank()`

d) `isPresent()`

15. Какой метод используется для повторения строки определенное количество раз?

a) `repeat()`

b) `duplicate()`

c) `replicate()`

d) `copy()`

16. Что вернет метод `lastIndexOf(substring)` для строки "Hello world, hello" при `substring = "hello"`?

a) 11

b) 12

c) 6

d) -1

17. Каким методом можно проверить, оканчивается ли строка на определенный суффикс?

- a) endsWith()
- b) isEndsWith()
- c) checkSuffix()
- d) hasSuffix()

18. Какой метод используется для преобразования строки в нижний регистр?

- a) toLowerCase()
- b) lowerCase()
- c) convertToLower()
- d) makeLowerCase()

19. Что вернет метод compareTo(otherString) для строки "java" при otherString = "java"?

- a) 0
- b) 1
- c) -1
- d) true

20. Каким методом можно развернуть строку наоборот?

- a) reverse()
- b) invert()
- c) flip()
- d) back()

Обработка символов.

Тест

1. Какой класс в Java используется для работы с символами?

- a) String
- b) Character
- c) Symbol
- d) Char

2. Каким методом можно получить длину строки в символах?

- a) length()
- b) size()
- c) count()
- d) getLength()

3. Какие символы в Java являются экранируемыми?

- a) @, !
- b) \$, #
- c) \, "
- d) &, *

4. Как объединить две строки в одну?

- a) concat()
- b) append()
- c) merge()
- d) combine()

5. Какой символ используется для обозначения конца строки?

- a) \n
- b) \t
- c) \r
- d) \b

6. Какой метод используется для преобразования символа в верхний регистр?
- a) toUpperCase()
 - b) toLowerCase()
 - c) capitalize()
 - d) changeCase()
7. Можно ли изменять символы в строке напрямую?
- a) Да, используя метод setCharAt()
 - b) Да, используя метод modifyChar()
 - c) Нет, строки иммутабельны
 - d) Нет, символы нельзя изменять
8. Какая функция позволяет проверить, является ли символ буквой алфавита?
- a) isLetter()
 - b) isAlphabetic()
 - c) isChar()
 - d) isAlpha()
9. Какое значение будет у символа '\u0031' после преобразования в строку?
- a) "1"
 - b) "\u0031"
 - c) "31"
 - d) "u0031"
10. Какие методы используются для разделения строки на отдельные символы?
- a) split()
 - b) divide()
 - c) separate()

d) toCharArray()

11. Какой символ используется для обозначения начала комментария в Java?

a) &

b) #

c) /

d) //

12. Какой метод позволяет проверить, является ли символ цифрой?

a) isDigit()

b) isNumber()

c) isInt()

d) isNumeric()

13. Какой символ используется для обозначения пустой строки?

a) ""

b) " "

c) Не существует спецсимвола для пустой строки

d) \0

14. Каким методом можно получить код символа в таблице Unicode?

a) toUnicode()

b) getUnicode()

c) codePointAt()

d) getCode()

15. Какой символ используется для обозначения конца файла?

a) \n

b) \r

c) \b

d) \0

16. Какой метод позволяет проверить, одинаковые ли символы в двух строках?

a) equals()

b) compare()

c) identical()

d) checkEquality()

17. Какой метод позволяет получить последний символ строки?

a) lastChar()

b) getLastChar()

c) charAt()

d) substring()

18. Каким методом можно заменить все вхождения одного символа на другой в строке?

a) replace()

b) swap()

c) substitute()

d) switchChar()

19. Какой символ используется для обозначения пробела?

a) " "

b) _

c) -

d) /

20. Какие методы используются для преобразования символа в численное значение?

- a) toNumeric()
- b) convertToNumber()
- c) intValue()
- d) parse()

Тема 5. Объектно-ориентированное программирование (ООП)

Тест

1. Что такое класс в Java?

- a) Определение объекта
- b) Набор методов и переменных
- c) Расширение другого класса

2. Как объявляется класс в Java?

- a) class MyClass
- b) MyClass = new class
- c) int MyClass = new int

3. Что такое конструктор класса?

- a) Метод, который вызывается при создании объекта класса
- b) Метод, который вызывается при удалении объекта класса
- c) Метод, который вызывается при изменении объекта класса

4. Может ли класс наследоваться от нескольких других классов в Java?

- a) Да
- b) Нет

5. Что такое метод в Java?

- a) Функция внутри класса
- b) Специальный тип переменной
- c) Описание объекта

6. Может ли метод иметь входные параметры?

- a) Да
- b) Нет

7. Как объявляется метод в Java?

- a) `void myMethod()`
- b) `myMethod = new void()`
- c) `myMethod = void`

8. Может ли метод возвращать значение?

- a) Да
- b) Нет

9. Что такое переменная в Java?

- a) Имя объекта
- b) Место для хранения данных
- c) Специальный тип данных

10. Как объявляется переменная в Java?

- a) `int myVar;`
- b) `myVar = new int();`
- c) `myVar = int;`

11. Можно ли объявить переменную без инициализации?

- a) Да

b) Нет

12. Может ли класс иметь статические переменные и методы?

a) Да

b) Нет

13. Что такое наследование в Java?

a) Возможность создания нового класса на основе существующего

b) Возможность изменения класса без изменения его исходного кода

c) Возможность использования переменной в разных классах

14. Что такое полиморфизм в Java?

a) Возможность использования одного имени для разных методов

b) Возможность использования разных имен для одного метода

c) Возможность использования одного метода для разных классов

15. Что такое инкапсуляция в Java?

a) Скрытие данных и методов внутри класса

b) Предоставление доступа к данным и методам других классов

c) Использование методов для изменения данных внутри класса

16. Что такое абстракция в Java?

a) Способ представления сложных объектов через их общие характеристики

b) Процесс создания экземпляра класса

c) Метод для определения типа данных переменной

17. Что такое интерфейс в Java?

a) Набор методов, которые должны быть реализованы классом

- b) Место для хранения данных класса
- c) Описание объекта класса

18. Может ли класс реализовать несколько интерфейсов в Java?

- a) Да
- b) Нет

19. Что такое поле класса (class field) в Java?

- a) Переменная, объявленная внутри класса
- b) Место для хранения методов класса
- c) Место для хранения экземпляров класса

20. Какие модификаторы доступа могут быть применены к классу в Java?

- a) public, private, protected
- b) static, final, abstract
- c) int, double, boolean

Разработка приложения в соответствии с принципами объектно-ориентированного программирования по индивидуальным заданиям (начальный этап).

Тест

1. Какие основные принципы объектно-ориентированного программирования (ООП) в Java?

- a) Наследование, инкапсуляция, полиморфизм
- b) Условные операторы, циклы, массивы
- c) Иерархия классов, наследование, интерфейсы

2. Что такое класс в Java?

- a) Шаблон или форма для создания объектов
- b) Переменная для хранения данных
- c) Подпрограмма, выполняющая определенную задачу

3. Что такое объект в Java?

- a) Экземпляр класса, имеющий свои собственные данные и методы
- b) Базовый элемент всех программ
- c) Численное значение, хранящееся в памяти

4. Что такое инкапсуляция в Java?

- a) Процесс объединения данных и методов, работающих с ними, в одном классе
- b) Процесс создания объектов на основе класса
- c) Возможность использования методов без создания объекта

5. Что представляют собой сеттеры и геттеры в Java?

- a) Методы для установки и получения значений частных переменных класса
- b) Методы для создания объектов
- c) Методы для выполнения арифметических операций

6. Что такое наследование в Java?

- a) Процесс создания нового класса на основе уже существующего
- b) Процесс повторного использования кода
- c) Процесс создания объектов на основе интерфейса

7. Что такое полиморфизм в Java?

- a) Возможность объектов разных классов использовать одинаковые методы
- b) Возможность одного объекта иметь разные значения
- c) Возможность использования нескольких различных типов данных

8. Что такое абстракция в Java?

- a) Процесс скрытия деталей реализации и предоставления общего интерфейса
- b) Процесс создания объектов на основе класса
- c) Процесс объединения данных и методов в одном классе

9. Что такое интерфейс в Java?

- a) Набор методов без реализации, которые должны быть определены в классе, реализующем этот интерфейс
- b) Объект, предоставляющий доступ к таблице значений
- c) Часть кода, выполняющая определенную задачу

10. Какие виды связей между классами могут быть в Java?

- a) Наследование, композиция, ассоциация
- b) Условные операторы, циклы, массивы
- c) Инкапсуляция, наследование, интерфейсы

11. Что такое перегрузка методов в Java?

- a) Возможность создания метода с одинаковым именем, но с разными параметрами
- b) Процесс создания нового класса на основе уже существующего
- c) Процесс создания объектов на основе интерфейса

12. Что такое переопределение методов в Java?

- a) Процесс создания новой реализации метода, уже определенного в родительском классе
- b) Процесс создания объектов на основе класса
- c) Процесс повторного использования кода

13. Какой модификатор доступа используется для класса с ограниченным доступом, доступного только внутри пакета?

- a) default

- b) public
- c) private

14. Какой модификатор доступа используется для метода или переменной, доступного только внутри класса?

- a) private
- b) protected
- c) public

15. Что такое абстрактный класс в Java?

- a) Класс, который не может быть создан, но может быть наследован
- b) Класс, который не имеет методов
- c) Класс, доступный только внутри пакета

16. Что такое перечисление (enum) в Java?

- a) Тип данных, ограниченный набором значений
- b) Класс, который не может быть создан, но может быть наследован
- c) Класс, который не имеет методов

17. Что такое композиция в Java?

- a) Связь между классами, когда один класс включает в себя объект другого класса
- b) Процесс создания нового класса на основе уже существующего
- c) Процесс повторного использования кода

18. Что такое статический метод в Java?

- a) Метод, который может быть вызван без создания объекта класса
- b) Метод, использующий только статические переменные
- c) Метод, использующий только приватные переменные

19. Что такое статическая переменная в Java?

- a) Переменная, общая для всех объектов класса
- b) Переменная, доступная только внутри класса
- c) Переменная, которая не может быть изменена после инициализации

20. Что такое исключение в Java?

- a) Ошибка, возникающая во время выполнения программы
- b) Отдельный тип данных, используемый для хранения значений
- c) Интерфейс для взаимодействия с пользователем

Тема 6. Потоки данных, работа с файловой системой

Обработка потоков в учебном проекте.

Тест

1. Что такое поток в Java?

- a) Участок программного кода
- b) Параллельно исполняемая последовательность инструкций
- c) Объект класса Thread

2. Какие классы используются для создания и управления потоками в Java?

- a) Thread и Runnable
- b) Process и Executor
- c) Stream и Buffer

3. Что такое многопоточность?

- a) Способность программы выполнять несколько потоков одновременно
- b) Возможность создания множества потоков
- c) Процесс увеличения производительности программы

4. Какой метод используется для создания нового потока в Java?
- a) start()
 - b) run()
 - c) create()
5. Что происходит, когда метод start() вызывается на объекте потока?
- a) Поток начинает выполняться
 - b) Методы run() и start() вызываются одновременно
 - c) Возникает исключение
6. Как можно создать поток в Java, реализуя интерфейс Runnable?
- a) Путем наследования от класса Thread
 - b) Реализовав метод run() интерфейса Runnable
 - c) Вызвав метод start() класса Runnable
7. Что такое синхронизация потоков?
- a) Организация выполнения потоков последовательно
 - b) Метод для остановки потоков
 - c) Управление приоритетом потоков
8. Как можно обеспечить синхронизацию потоков в Java?
- a) Используя ключевое слово synchronized
 - b) Назначая разные приоритеты потокам
 - c) Создавая дополнительные экземпляры классов Thread
9. Какой класс в Java используется для работы с блокировками?
- a) Lock
 - b) Sync
 - c) Mutex

10. Что такое мьютекс?

- a) Объект для синхронизации потоков
- b) Поток, который выполняется в единственном экземпляре
- c) Класс для работы с многопоточностью

11. Какие методы реализуют блокировки в интерфейсе Lock?

- a) lock() и unlock()
- b) wait() и notify()
- c) start() и stop()

12. Что означает метод notifyAll() в Java?

- a) Уведомляет все потоки, ожидающие на объекте
- b) Останавливает все потоки
- c) Запускает новый поток

13. Что такое межпоточное взаимодействие?

- a) Способность нескольких потоков взаимодействовать друг с другом
- b) Автономная работа каждого потока
- c) Управление приоритетом потоков

14. Какие методы используются для межпоточного взаимодействия?

- a) wait() и notify()
- b) start() и stop()
- c) execute() и submit()

15. Для чего используется метод yield() в Java?

- a) Поток освобождает процессор для других потоков
- b) Поток увеличивает свой приоритет

с) Поток останавливается и ждет уведомления

16. Какие состояния может иметь поток в Java?

- а) Новый, готовый, блокированный, исполняемый, завершенный
- б) Активный, пассивный, блокированный, ожидающий, завершенный
- с) Запущенный, приостановленный, завершенный

17. Как можно приостановить выполнение потока в Java?

- а) Методом `sleep()`
- б) Используя ключевое слово `pause`
- с) Вызвав метод `stop()`

18. Для чего используется метод `interrupt()` в Java?

- а) Для завершения потока
- б) Для отправки сигнала потоку о необходимости завершить выполнение
- с) Для перевода потока в режим ожидания

19. Что такое демон-поток в Java?

- а) Поток, который запускается автоматически при запуске программы
- б) Поток, который выполняется на заднем плане и завершается при завершении программы
- с) Поток, который имеет высокий приоритет

20. Как можно обеспечить безопасную работу с разделяемыми ресурсами для потоков?

- а) Используя мьютексы
- б) Применяя мониторы
- с) Используя ключевое слово `volatile`

21. Что такое внутренний класс потока в Java?

- a) Класс, наследуемый от класса Thread
- b) Класс, реализующий интерфейс Runnable
- c) Класс, объявленный внутри другого класса

22. Как можно получить текущий поток выполнения в Java?

- a) С помощью статического метода `currentThread()` класса Thread
- b) Используя метод `send()` класса Thread
- c) Через объект класса Runnable

23. Какие методы используются для установки и получения приоритета потока?

- a) `setPriority()` и `getPriority()`
- b) `start()` и `stop()`
- c) `execute()` и `submit()`

24. Как можно узнать, выполняется ли поток в данный момент?

- a) С помощью метода `isAlive()`
- b) Через метод `yield()`
- c) Используя метод `join()`

25. Какие методы класса Object используются для работы с мониторами?

- a) `wait()`, `notify()` и `notifyAll()`
- b) `lock()` и `unlock()`
- c) `setMonitor()` и `getMonitor()`

Обработка файлов в учебном проекте.

Тест

1. Какой класс в Java используется для работы с файлами?

- a) FileReader
- b) FileWriter
- c) File
- d) FileInputStream

2. Какая библиотека Java предоставляет классы для работы с файлами?

- a) java.io
- b) java.nio
- c) java.util
- d) java.lang

3. Какой метод используется для чтения строки из файла в Java?

- a) readLine()
- b) readString()
- c) read()
- d) getNextLine()

4. Какой метод используется для записи строки в файл в Java?

- a) writeLine()
- b) writeString()
- c) write()
- d) appendLine()

5. Какой класс используется для работы с байтами файлов в Java?

- a) BufferedWriter
- b) BufferedReader
- c) ByteArrayOutputStream
- d) FileOutputStream

6. Какой метод используется для проверки существования файла в Java?
- a) exists()
 - b) isExist()
 - c) isFileExist()
 - d) fileExists()
7. Какой метод используется для создания нового файла в Java?
- a) create()
 - b) newFile()
 - c) makeFile()
 - d) createNewFile()
8. Какой метод используется для удаления файла в Java?
- a) remove()
 - b) delete()
 - c) erase()
 - d) deleteFile()
9. Какой метод используется для переименования файла в Java?
- a) rename()
 - b) renameFile()
 - c) move()
 - d) renameTo()
10. Какой класс используется для чтения и записи объектов в файлы в Java?
- a) ObjectStream
 - b) ObjectOutputStream
 - c) ObjectFileReader
 - d) ObjectWriter

11. Какой метод используется для закрытия файла в Java?

- a) `closeFile()`
- b) `close()`
- c) `finish()`
- d) `exit()`

12. Какой метод используется для получения списка файлов в директории в Java?

- a) `getFiles()`
- b) `listFiles()`
- c) `readFiles()`
- d) `fetchFiles()`

13. Какой класс используется для создания временного файла в Java?

- a) `TempFile`
- b) `TempFileManager`
- c) `TempFileWriter`
- d) `File.createTempFile()`

14. Какой метод используется для указания пути к файлу в Java?

- a) `setPath()`
- b) `setFilePath()`
- c) `setFile()`
- d) `setAbsolutePath()`

15. Какой метод используется для получения размера файла в байтах в Java?

- a) `getSize()`
- b) `getLength()`

- c) length()
- d) getFileSize()

16. Какой метод используется для чтения содержимого файла в массив байтов в Java?

- a) readBytes()
- b) getBytes()
- c) toArray()
- d) readAllBytes()

17. Какой метод используется для записи массива байтов в файл в Java?

- a) bytesToFile()
- b) writeBytes()
- c) toFile()
- d) writeAllBytes()

18. Какой метод используется для проверки, является ли файл директорией в Java?

- a) isDirectory()
- b) isFolder()
- c) isDir()
- d) isDirectoryFile()

19. Какой метод используется для получения последнего времени модификации файла в Java?

- a) getLastModified()
- b) getModifiedDate()
- c) getModifiedTime()
- d) getModificationDate()

20. Какой метод используется для получения списка файлов в директории, включая поддиректории в Java?

- a) `getFilesWithSubdirectories()`
- b) `listFilesRecursively()`
- c) `readFilesWithSubdirectories()`
- d) `fetchFilesRecursively()`

21. Какая аннотация используется для обработки исключений при работе с файлами в Java?

- a) `@TryCatch`
- b) `@Catch`
- c) `@ExceptionHandler`
- d) `@ExceptionHandling`

22. Какой метод используется для копирования файла в Java?

- a) `copyFile()`
- b) `copy()`
- c) `copyTo()`
- d) `copyFileTo()`

Доработка приложения с учетом обработки файлов и потоков.

Тест

1. Каким образом можно открыть текстовый файл в Java?

- a) Используя класс `Scanner`
- b) Используя класс `BufferedReader`
- c) Используя класс `FileReader`
- d) Все перечисленные варианты

2. Какой класс в Java используется для записи данных в файл?

- a) `PrintWriter`

- b) `BufferedWriter`
- c) `FileWriter`
- d) Все перечисленные варианты

3. Каким образом можно прочитать содержимое файла и сохранить его в виде строки?

- a) Используя метод `read()` из класса `FileReader`
- b) Используя метод `nextLine()` из класса `Scanner`
- c) Используя метод `readLine()` из класса `BufferedReader`
- d) Все перечисленные варианты

4. Каким методом из класса `PrintWriter` можно записать строку в файл?

- a) `writeString()`
- b) `append()`
- c) `print()`
- d) Все перечисленные варианты

5. Какая из перечисленных операций позволяет удалить файл в Java?

- a) `delete()`
- b) `remove()`
- c) `erase()`
- d) Ни один из предложенных вариантов

6. Какой класс в Java используется для работы с двоичными данными?

- a) `BinaryReader`
- b) `BinaryWriter`
- c) `DataInputStream`
- d) Все перечисленные варианты

7. Как можно прочитать двоичные данные из файла, используя класс `DataInputStream`?

- a) Используя метод `readByte()`
- b) Используя метод `readBoolean()`
- c) Используя метод `readInt()`
- d) Все перечисленные варианты

8. Как можно записать двоичные данные в файл, используя класс `DataOutputStream`?

- a) Используя метод `writeByte()`
- b) Используя метод `writeBoolean()`
- c) Используя метод `writeInt()`
- d) Все перечисленные варианты

9. Что делает метод `flush()` для потока вывода?

- a) Сбрасывает буфер и записывает данные в выходной поток
- b) Закрывает поток
- c) Очищает буфер для ввода данных
- d) Переключает поток на чтение

10. Какой интерфейс в Java предоставляет функциональность для обработки потоков?

- a) `Stream`
- b) `Reader`
- c) `Writer`
- d) `InputStream`

11. Какую функцию выполняет класс `InputStreamReader`?

- a) Преобразует поток байтов в поток символов
- b) Выполняет чтение данных из файла

- c) Позволяет записывать данные в файл
- d) Все перечисленные варианты

12. Какой метод класса `InputStreamReader` используется для чтения одного символа?

- a) `read()`
- b) `readChar()`
- c) `readSymbol()`
- d) Все перечисленные варианты

13. Какой метод класса `BufferedReader` используется для чтения строки из потока?

- a) `readString()`
- b) `readLine()`
- c) `readText()`
- d) Все перечисленные варианты

14. Какой метод класса `OutputStreamWriter` используется для записи одного символа?

- a) `write()`
- b) `writeChar()`
- c) `writeSymbol()`
- d) Все перечисленные варианты

15. Какой метод класса `BufferedWriter` используется для записи строки в поток?

- a) `writeString()`
- b) `writeLine()`
- c) `writeText()`
- d) Все перечисленные варианты

Тема 7. Коллекции и интерфейсы

Использование коллекций в учебном проекте.

Тест

1. Что такое коллекция в Java?
 - a) Массив объектов
 - b) Группа объектов
 - c) Структура данных
 - d) Все вышеперечисленное

2. Какие коллекции в Java поддерживают упорядоченное хранение элементов?
 - a) List
 - b) Set
 - c) Map
 - d) Все вышеперечисленные

3. Какое основное преимущество использования коллекций в Java?
 - a) Увеличение производительности
 - b) Повышение безопасности
 - c) Упрощение кода
 - d) Расширение возможностей языка

4. Какие коллекции в Java не позволяют хранить дублирующиеся элементы?
 - a) List
 - b) Set

c) Queue

d) Map

5. Какой метод используется для добавления элемента в коллекцию?

a) add()

b) insert()

c) push()

d) append()

6. Какой метод используется для удаления элемента из коллекции?

a) remove()

b) delete()

c) pop()

d) erase()

7. Какие коллекции в Java поддерживают использование индексов для доступа к элементам?

a) Set

b) Map

c) List

d) Queue

8. Какой метод используется для получения размера коллекции?

a) size()

b) length()

c) count()

d) getLength()

9. Какой метод используется для проверки наличия элемента в коллекции?
- a) contains()
 - b) exists()
 - c) containsElement()
 - d) has()
10. Какие коллекции в Java могут быть отсортированы по умолчанию?
- a) List
 - b) Set
 - c) Map
 - d) Все вышеперечисленные
11. Какой метод используется для сортировки коллекции?
- a) sort()
 - b) order()
 - c) organize()
 - d) arrange()
12. Какой интерфейс является базовым для всех коллекций в Java?
- a) Collection
 - b) List
 - c) Map
 - d) Iterable
13. Какие коллекции в Java могут содержать ключи и значения?
- a) List
 - b) Set

- c) Map
- d) Queue

14. Какой метод используется для получения элемента коллекции по индексу?

- a) get()
- b) retrieve()
- c) fetch()
- d) obtain()

15. Какой метод используется для обхода элементов коллекции?

- a) foreach()
- b) for()
- c) iterate()
- d) loop()

16. Какой метод используется для объединения двух коллекций?

- a) merge()
- b) combine()
- c) join()
- d) addAll()

17. Какой метод используется для поиска элемента в коллекции по заданному условию?

- a) find()
- b) search()
- c) match()
- d) filter()

18. Какой метод используется для получения подколлекции заданного диапазона элементов?

- a) subList()
- b) range()
- c) slice()
- d) subset()

19. Какой метод используется для очистки коллекции от всех элементов?

- a) clear()
- b) empty()
- c) erase()
- d) remove()

20. Какой метод используется для получения первого элемента коллекции?

- a) getFirst()
- b) first()
- c) head()
- d) element()

21. Какой метод используется для получения последнего элемента коллекции?

- a) getLast()
- b) last()
- c) tail()
- d) end()

Реализация параметризованного интерфейса в учебном проекте.

Тест

1. Какое ключевое слово используется для определения параметризованного интерфейса в Java?
 - a) generic
 - b) interface
 - c) implements

2. Как объявить параметризованный интерфейс?
 - a) `public interface MyInterface<T>`
 - b) `public interface MyInterface`
 - c) `public interface T extends MyInterface`

3. Что такое тип-параметр в параметризованном интерфейсе?
 - a) Это значимый тип данных в языке Java
 - b) Это класс или интерфейс, указанный в угловых скобках после имени интерфейса
 - c) Это название переменной, которую можно использовать внутри интерфейса

4. Какой оператор используется для указания обобщенного типа в параметризованном интерфейсе?
 - a) `<`
 - b) `>`
 - c) `[`

5. Можно ли использовать несколько типов-параметров в одном параметризованном интерфейсе?
 - a) Да
 - b) Нет

6. Какой тип-параметр используется в следующем объявлении интерфейса: `MyInterface<T, U>`?

- a) T
- b) U
- c) Оба типа T и U

7. Может ли класс реализовать несколько параметризованных интерфейсов с разными типами-параметрами?

- a) Да
- b) Нет

8. Какие операции можно выполнять с типом-параметром в параметризованном интерфейсе?

- a) Создание экземпляров класса
- b) Вызов методов класса
- c) Оба варианта

9. Может ли параметризованный интерфейс иметь статические методы?

- a) Да
- b) Нет

10. Какой модификатор доступа используется при объявлении методов в параметризованном интерфейсе?

- a) `public`
- b) `private`
- c) `default`

11. Можно ли создавать экземпляры параметризованного интерфейса?

- a) Да

b) Нет

12. Какой оператор используется для параметризации класса с использованием параметризованного интерфейса?

a) implements

b) extends

c) implements и extends

13. Как объявить класс, который реализует параметризованный интерфейс?

a) `public class MyClass<T> implements MyInterface<T>`

b) `public class MyClass implements MyInterface`

14. Можно ли использовать параметризованный интерфейс без указания его типа-параметра?

a) Да

b) Нет

15. Каким образом класс реализует методы параметризованного интерфейса с использованием конкретных типов?

a) Путем указания типа-параметра в угловых скобках при реализации интерфейса

b) Автоматически, при реализации интерфейса

16. Что произойдет, если класс реализует параметризованный интерфейс, но не указывает его тип-параметр?

a) Ошибка компиляции

b) Предупреждение компилятора

17. Какой тип-параметр используется в следующем объявлении интерфейса: `MyInterface<T extends Number>`?

a) T

b) Number

18. Может ли класс расширять другой класс и реализовывать параметризованный интерфейс одновременно?

a) Да

b) Нет

Тема 8. Разработка интерфейса пользователя

Создание форм.

Тест

1. Что такое GUI?

a) Graphical User Interface (графический пользовательский интерфейс)

b) General User Interface (общий пользовательский интерфейс)

c) Graphical User Innovation (инновация графического интерфейса пользователя)

2. Какая библиотека в Java используется для создания форм?

a) javax.swing

b) java.form

c) javax.forms

3. Какой класс в javax.swing используется для создания главного окна приложения?

a) JFrame

b) JPanel

c) JWindow

4. Как добавить компонент на форму в Java?

a) form.add(component);

b) form.addComponent(component);

c) form.getContentPane().add(component);

5. Какой класс используется для создания метки (текстового поля) в Java?

- a) JLabel
- b) JTextField
- c) JTextArea

6. Как добавить кнопку на форму в Java?

- a) form.add(button);
- b) form.addButton(button);
- c) form.getContentPane().add(button);

7. Какой метод используется для обработки события при нажатии кнопки?

- a) addActionListener()
- b) addClickListener()
- c) addButtonListener()

8. Какой класс используется для создания выпадающего списка в Java?

- a) JComboBox
- b) JList
- c) JDropDownList

9. Как добавить выпадающий список на форму в Java?

- a) form.add(dropDown);
- b) form.addComboBox(dropDown);
- c) form.getContentPane().add(dropDown);

10. Какой класс используется для создания флажков в Java?

- a) JCheckBox
- b) JRadioButton

c) JToggleButton

11. Как добавить флажок на форму в Java?

- a) form.add(checkBox);
- b) form.addBox(checkBox);
- c) form.getContentPane().add(checkBox);

12. Какой класс используется для создания радиокнопок в Java?

- a) JRadioButton
- b) JCheckBox
- c) JButton

13. Как добавить радиокнопку на форму в Java?

- a) form.add(radioButton);
- b) form.addRadio(radioButton);
- c) form.getContentPane().add(radioButton);

14. Какой класс используется для создания текстового поля в Java?

- a) JTextField
- b) JLabel
- c) JTextArea

15. Как добавить текстовое поле на форму в Java?

- a) form.add(textField);
- b) form.addTextField(textField);
- c) form.getContentPane().add(textField);

16. Каким методом можно задать текст по умолчанию для текстового поля?

- a) setDefaultText()

- b) `setPlaceholder()`
- c) `setText()`

17. Какой класс используется для создания ползунка в Java?

- a) `JSlider`
- b) `JScrollBar`
- c) `JProgress`

18. Как добавить ползунок на форму в Java?

- a) `form.add(slider);`
- b) `form.addSlider(slider);`
- c) `form.getContentPane().add(slider);`

19. Каким методом задается начальное значение ползунка в Java?

- a) `setValue()`
- b) `setMinValue()`
- c) `setInitialValue()`

20. Какой класс используется для создания окна сообщения в Java?

- a) `JOptionPane`
- b) `JDialog`
- c) `JMessage`

21. Как показать окно сообщения в Java?

- a) `JOptionPane.show()`
- b) `JOptionPane.showMessageDialog()`
- c) `JOptionPane.showMessageDialogDialog()`

22. Как задать название для окна сообщения в Java?

- a) setTitle()
- b) setMessage()
- c) setHeaderText()

23. Какой класс используется для создания файлового выбора в Java?

- a) JFileChooser
- b) JFileSelector
- c) JFilePicker

24. Как показать файловый выбор на форме в Java?

- a) form.add(fileChooser);
- b) form.addFilePicker(fileChooser);
- c) form.getContentPane().add(fileChooser);

25. Какой метод используется для получения выбранного файла с файлового выбора?

- a) getSelectedFile()
- b) getFile()
- c) selectFile()

26. Какой класс используется для создания календаря в Java?

- a) JCalendar
- b) JDateChooser
- c) JPicker

Добавление кнопок, меток, текстовых полей.

Тест

1. Что такое кнопка в Java?

- a) Элемент управления, с помощью которого пользователь может осуществлять действия

b) Наглядное отображение текста или графики

2. Как добавить кнопку на панель в Java?

a) Создать новый экземпляр класса JButton и добавить его на панель

b) Нажать кнопку "Добавить", затем выбрать "Кнопка" из списка

3. Какой метод используется для установки текста на кнопку?

a) setText()

b) setLabel()

4. Как добавить метку (label) на панель в Java?

a) Создать новый экземпляр класса JLabel и добавить его на панель

b) Нажать кнопку "Добавить", затем выбрать "Метка" из списка

5. Как установить текст для метки?

a) Использовать метод setText()

b) Использовать метод setLabel()

6. Как добавить текстовое поле (text field) на панель в Java?

a) Создать новый экземпляр класса JTextField и добавить его на панель

b) Нажать кнопку "Добавить", затем выбрать "Текстовое поле" из списка

7. Как получить текст из текстового поля?

a) Использовать метод getText()

b) Использовать метод setText()

8. Как установить ширина текстового поля?

a) Использовать метод setPreferredSize()

b) Использовать метод setWidth()

9. Что такое обработчик событий (event handler)?

- a) Код, выполняющийся при возникновении определенного события
- b) Кнопка, при нажатии на которую возникает событие

10. Как называется метод, который добавляет обработчик событий к кнопке?

- a) addActionListener()
- b) addEventHandler()

11. Как добавить действие к кнопке, которое будет выполняться при ее нажатии?

- a) Создать новый экземпляр класса ActionListener и добавить его к кнопке
- b) Нажать кнопку "Добавить действие", затем выбрать нужное действие из списка

12. Что делает метод actionPerformed() обработчика событий?

- a) Описывает действия, которые нужно выполнить при возникновении события
- b) Проверяет, была ли нажата кнопка

13. Какие значения может принимать параметр в методе addActionListener()?

- a) Объект класса, реализующего интерфейс ActionListener
- b) Строка с кодом, который нужно выполнить при нажатии на кнопку

14. Как задать расположение кнопок, меток и текстовых полей на панели?

- a) Использовать менеджер компоновки (layout manager)
- b) Вручную задавать координаты каждого элемента

15. Как называется менеджер компоновки, который располагает элементы в одной строке?

- a) FlowLayout
- b) GridLayout

16. Какая разметка используется в менеджере компоновки GridLayout?

- a) Таблица с равномерно распределенными ячейками
- b) Строки и столбцы с фиксированными размерами

17. Какая разметка используется в менеджере компоновки FlowLayout?

- a) Элементы выравниваются в одной строке, идущей друг за другом
- b) Элементы размещаются в таблице с равномерно распределенными ячейками

18. Какая разметка используется в менеджере компоновки BorderLayout?

- a) Элементы размещаются на севере, юге, западе и востоке контейнера
- b) Элементы выравниваются в одной строке

Интерфейс формы и размещение компонентов.

Тест

1. Что такое интерфейс формы в Java?

- a. Метод, позволяющий создать новую форму
- b. Инструмент, позволяющий размещать компоненты на форме
- c. Класс, представляющий пользовательский интерфейс

2. Какой класс используется для создания формы в Java?

- a. JFrame
- b. JButton
- c. JPasswordField

3. Какие компоненты можно размещать на форме?

- a. Только текстовые поля
 - b. Только кнопки
 - c. Различные компоненты, такие как кнопки, текстовые поля и другие
4. Какие методы используются для установки расположения компонентов на форме?
- a. `setLayout()`
 - b. `setPosition()`
 - c. `setLocation()`
5. Какое расположение компонентов является наиболее популярным и удобным для работы с формами?
- a. `FlowLayout`
 - b. `GridLayout`
 - c. `BorderLayout`
6. Какие методы используются для добавления компонентов на форму?
- a. `add()`
 - b. `insert()`
 - c. `place()`
7. Какой метод позволяет задать размер компонента на форме?
- a. `setSize()`
 - b. `setWidth()`
 - c. `setDimension()`
8. Как задать заголовок формы?
- a. `setTitle()`
 - b. `setHeader()`
 - c. `setCaption()`

9. Какой метод вызывается для отображения формы на экране?
- a. setVisible()
 - b. show()
 - c. display()
10. Какой метод вызывается при закрытии формы?
- a. onClose()
 - b. onExit()
 - c. setDefaultCloseOperation()
11. Что такое слушатели событий в контексте интерфейса формы?
- a. Компоненты, отвечающие за обработку действий пользователя
 - b. Классы, отслеживающие события, происходящие на форме
 - c. Методы, определяющие взаимодействие между компонентами
12. Какой метод используется для добавления слушателя событий к компоненту?
- a. addActionListener()
 - b. addEventListener()
 - c. addListener()
13. Какой метод вызывается при возникновении события на компоненте?
- a. eventOccurred()
 - b. actionPerformed()
 - c. eventTriggered()
14. Какой класс позволяет создать модальное окно в Java?
- a. JDialog

- b. JFrame
- c. JWindow

15. Какое расположение компонентов позволяет создать сетку с заданным количеством строк и столбцов?

- a. GridLayout
- b. BorderLayout
- c. BoxLayout

16. Какое расположение компонентов позволяет размещать их по горизонтали или вертикали

- a. FlowLayout
- b. GridLayout
- c. BorderLayout

17. Какой класс используется для создания текстового поля в Java?

- a. JTextField
- b. JLabel
- c. JTextArea

18. Какой класс используется для создания кнопки в Java?

- a. JButton
- b. JTextField
- c. JCheckBox

19. Какой класс используется для создания метки в Java?

- a. JLabel
- b. JTextField
- c. JButton

20. Какой класс используется для создания флажка (checkbox) в Java?

- a. JCheckBox
- b. JTextField
- c. JRadioButton

21. Какой класс используется для создания радиокнопки (radiobutton) в Java?

- a. JRadioButton
- b. JTextField
- c. JCheckBox

Разработка кода обработки событий в учебном проекте.

Тест

1. Какие методы отвечают за обработку событий в Java?

- a) start() и stop()
- b) receiveEvent() и handleEvent()
- c) addEventListener() и actionPerformed()

2. Какой интерфейс необходимо реализовать для обработки событий мыши?

- a) ActionListener
- b) MouseListener
- c) KeyListener

3. Какой класс отвечает за обработку событий нажатия клавиш на клавиатуре?

- a) MouseAdapter
- b) KeyAdapter
- c) ActionListener

4. Какой метод вызывается при обработке события нажатия на кнопку?

- a) actionPerformed()
- b) mouseClicked()
- c) keyPressed()

5. Что делает метод actionPerformed()?

- a) Изменяет цвет фона окна
- b) Выполняет пользовательский код при возникновении события
- c) Открывает новое окно программы

6. Какой метод вызывается при обработке события движения мыши?

- a) mousePressed()
- b) mouseMoved()
- c) mouseDragged()

7. Какой метод вызывается при обработке события наведения курсора на компонент?

- a) mouseEntered()
- b) mouseExited()
- c) mouseMoved()

8. Какой класс отвечает за обработку событий изменения размеров окна?

- a) WindowAdapter
- b) ResizeAdapter
- c) SizeListener

9. Что делает метод mouseExited()?

- a) Изменяет цвет фона окна
- b) Проверяет, наведен ли курсор на компонент
- c) Выполняет пользовательский код при выходе курсора из компонента

10. Какой метод вызывается при обработке события изменения размеров окна?

- a) windowOpened()
- b) windowResized()
- c) componentResized()

11. Что делает метод mouseClicked()?

- a) Запускает процесс обработки событий нажатия клавиш
- b) Выполняет пользовательский код при клике мышью
- c) Меняет положение компонента на экране

12. Какой класс отвечает за обработку событий фокуса компонента?

- a) FocusAdapter
- b) KeyAdapter
- c) MouseAdapter

13. Какой метод вызывается при обработке события получения фокуса компонентом?

- a) focusLost()
- b) focusGained()
- c) focusChanged()

14. Что делает метод keyTyped()?

- a) Изменяет размер компонента
- b) Проверяет, нажаты ли определенные клавиши
- c) Выполняет пользовательский код при вводе текста

15. Какой метод вызывается при обработке события потери фокуса компонентом?

- a) focusGained()
- b) focusLost()
- c) focusChanged()

16. Что делает метод keyPressed()?

- a) Изменяет цвет фона окна
- b) Выполняет пользовательский код при нажатии клавиши
- c) Открывает новое окно программы

17. Какой интерфейс необходимо реализовать для обработки событий клавиатуры?

- a) ActionListener
- b) KeyListener
- c) MouseListener

18. Какой метод вызывается при обработке события перемещения мыши с зажатой кнопкой?

- a) mouseDragged()
- b) mousePressed()
- c) mouseMoved()

19. Какие классы отвечают за обработку событий ввода текста?

- a) ActionListener и MouseAdapter
- b) FocusAdapter и KeyAdapter
- c) WindowAdapter и ResizeAdapter

Тема 9. Платформа Android. Особенности программирования в Android Studio.

Проверочная работа

1. Что такое Android Studio и для чего он используется?
2. Какой язык программирования используется для разработки приложений в Android Studio?
3. Какие основные компоненты разработки приложений в Android Studio?
4. Что такое макеты (Layouts) в Android Studio и как они используются?
5. Как создать новый проект в Android Studio?
6. Как добавить новый экран (Activity) в проект Android Studio?
7. Какие инструменты для отладки и тестирования приложений предоставляет Android Studio?
8. Каким образом можно добавить элементы пользовательского интерфейса, такие как кнопки и текстовые поля, в макет приложения?
9. Как провести тестирование разработанного приложения на эмуляторе устройства в Android Studio?
10. Какие основные шаги для публикации приложения в магазине Google Play?

Практическая работа № 21. Модификация учебного проекта в Android Studio.

Проверочная работа

1. Что такое модификация учебного проекта в Android Studio?
2. Какими способами можно модифицировать учебный проект?
3. Как добавить новый экран (Activity) в учебный проект?

4. Как изменить внешний вид элемента интерфейса (View) в учебном проекте?
5. Как добавить новый элемент интерфейса в учебный проект?
6. Как изменить текстовое содержимое элемента интерфейса в учебном проекте?
7. Как добавить действие (Action) при нажатии на кнопку в учебном проекте?
8. Как передать данные между экранами (Activities) в учебном проекте?
9. Как добавить возможность работы с базой данных в учебный проект?
10. Как добавить возможность работы с сетью (API) в учебный проект?
11. Как изменить цвет фона учебного проекта?
12. Как добавить изображение в учебный проект?
13. Как изменить портретную ориентацию учебного проекта на альбомную?
14. Как настроить переход между экранами (Activities) с анимацией в учебном проекте?
15. Как установить учебный проект на реальное Android устройство?

ИЛИ

Часть 1: Теоретическая часть (10 баллов)

1. Кратко опишите процесс разработки Android-приложения.
2. Что такое AndroidManifest.xml и какая роль этого файла в разработке Android-приложения?
3. Какие основные компоненты Android-приложения вы знаете? Опишите каждый из них.
4. Что такое активности (Activity) и какие методы жизненного цикла они имеют?
5. Что такое макеты (Layouts) в Android? Опишите различные типы макетов.

Часть 2: Практическая часть (20 баллов)

Вам предлагается модифицировать учебный проект в Android Studio "TODO List" (список дел) для добавления новой функциональности.

1. Добавьте новый экран в приложение "TODO List", на который пользователь будет попадать при нажатии на каждый элемент списка задач. Назовите его "TaskDetailsActivity". (5 баллов)
2. В экране "TaskDetailsActivity" отобразите подробную информацию о выбранной задаче, такую как название, описание, дата создания и статус выполнения. (5 баллов)
3. Добавьте возможность редактирования задачи в экране "TaskDetailsActivity". При нажатии на кнопку "Редактировать" пользователь должен переходить на новый экран "EditTaskActivity", где он сможет изменять параметры задачи. (5 баллов)
4. В экране "EditTaskActivity" добавьте поле ввода для изменения названия задачи. Реализуйте сохранение изменений и вывод обновленной информации в экране "TaskDetailsActivity". (5 баллов)

Часть 3: Бонусные задания (дополнительные 5 баллов)

1. Добавьте возможность удалять задачи из списка. Реализуйте удаление задачи по долгому нажатию на элемент списка в экране "MainActivity".
2. Добавьте возможность сортировать задачи по дате создания или статусу выполнения в экране "MainActivity". Реализуйте переключатель или выпадающий список для выбора сортировки.

Практическая работа № 22. Разработка меню в учебном проекте.

Проверочная работа

1. Что такое меню в Android Studio?
2. Какие виды меню существуют в Android Studio?
3. Как создать опции меню в проекте?
4. Как добавить иконку к опции меню?
5. Как задать обработчик события при выборе опции меню?
6. Как добавить подменю в проект?
7. Каким образом можно задать группу опций в меню?

8. Каким способом можно скрыть или отобразить опцию меню в процессе работы приложения?
9. Каким образом можно создать контекстное меню в Android Studio?
10. Что такое Navigation Drawer в проекте Android Studio и как его добавить?
11. Как добавить Navigation Drawer в проект?
12. Как задать обработчик событий для элементов Navigation Drawer?
13. Каким образом можно скрыть или отобразить элементы Navigation Drawer в процессе работы приложения?
14. Как создать адаптивное меню, адаптированное для разных разрешений экрана?
15. Каким образом можно изменить внешний вид меню с помощью стилей и тем?