**Вопросы и ответы для башни Java**

**1. Введение в Java**

**1.1 Что такое Java**

**Что такое Java?**

a) Язык программирования, разработанный компанией Microsoft.

b) Язык программирования, созданный в 1996 году компанией Oracle.

c) Одно из наиболее распространенных языков программирования, используемых для создания мобильных приложений.

d) Язык программирования, который требует перекомпиляции для каждой платформы.

**Какая особенность языка Java делает его кроссплатформенным?**

a) Трансляция кода в байт-код и выполнение на виртуальной машине JVM.

b) Компиляция кода в машинный код каждой платформы отдельно.

c) Интерпретация кода на каждой платформе непосредственно при выполнении.

d) Использование только для операционных систем семейства Windows.

**Какие области использования включает в себя Java?**

a) Только разработка веб-приложений.

b) Разработка только для мобильных устройств.

c) Работа с базами данных, игровая разработка, исследовательские приложения и встроенное программирование.

d) Использование Java ограничено созданием научных приложений.

**Что такое JVM (Java Virtual Machine)?**

a) Виртуальная машина для выполнения JavaScript кода.

b) Программа, которая интерпретирует Java код в машинный код каждой платформы.

c) Среда выполнения, которая обеспечивает кроссплатформенность Java приложений.

d) Виртуальное устройство для отладки Java кода.

**Какие преимущества обеспечивает автоматическая сборка мусора в Java?**

a) Позволяет сборщику мусора самостоятельно выбирать, какие объекты удалять.

b) Уменьшает необходимость вручную освобождать память от неиспользуемых объектов.

c) Значительно увеличивает объем оперативной памяти, используемой приложением.

d) Автоматическая сборка мусора отсутствует в языке Java.

**Для чего используется JDBC в Java?**

a) Для создания графических десктопных приложений.

b) Для взаимодействия с реляционными базами данных.

c) Для разработки мобильных приложений под Android.

d) Для работы с виртуальной машиной JVM.

**Что представляет собой JavaFX?**

a) Фреймворк для создания игр.

b) Среда выполнения для Java приложений.

c) Интерпретатор Java кода.

d) Библиотека для разработки графических десктопных приложений.

**Какие операционные системы поддерживают выполнение Java приложений?**

a) Только Windows.

b) Только Linux.

c) Windows, Linux, Mac OS и другие.

d) Только Mac OS.

**Какой язык программирования тесно связан с Java с точки зрения синтаксиса?**

a) Python.

b) C++.

c) JavaScript.

d) Ruby.

**2.** **Синтаксис Java**

**2.1** **Переменные в Java**

**Что представляет собой переменная в Java?**

a) Определенная область кода, где хранятся данные.

b) Инструкция для выполнения определенной операции.

c) Именованная область памяти, которая хранит значение определенного типа.

d) Функция, используемая для создания объектов.

**Какой символ используется для присваивания значения переменной в Java?**

a) ==

b) =

c) ->

d) :

**Какие требования к именам переменных в Java?**

a) Имя может содержать любые символы, включая пробелы.

b) Первый символ в имени должен быть цифрой.

c) В имени не должно быть знаков пунктуации, кроме подчеркивания.

d) Имя не может быть ключевым словом языка Java.

**Что означает выражение "регистрозависимый язык" в контексте переменных в Java?**

a) Имена переменных должны быть написаны капсом.

b) В Java можно использовать только строчные буквы для имен переменных.

c) Java различает заглавные и строчные буквы в именах переменных.

d) Java игнорирует регистр при определении имен переменных.

**Какая типовая форма записи для объявления переменной с последующим присвоением значения в Java?**

a) int x;

b) x = int 9;

c) int x = 9;

d) x 9;

**Какой тип данных определяется ключевым словом "int" в Java?**

a) Символьный тип данных.

b) Логический тип данных.

c) Целочисленный тип данных.

d) Вещественный тип данных.

**Какой символ используется для окончания оператора в Java?**

a) :

b) .

c) ,

d) ;

**Какой будет значение переменной, если оно не было явно задано?**

a) Оно будет равно 1.

b) Оно будет равно 0.

c) Оно будет равно null.

d) Оно будет зависеть от типа переменной.

**Какой стиль написания имен переменных рекомендуется в Java?**

a) Имена должны быть написаны капсом.

b) Имена должны содержать только строчные буквы.

c) Имена должны быть вида CamelCase.

d) Имена должны содержать пробелы между словами.

**2.2** **Подробнее про целочисленные типы данных**

**Какой размер (в битах) имеет тип данных byte в Java?**

a) 16 бит

b) 32 бита

c) 8 бит (1 байт)

d) 64 бита

**Какой диапазон значений для типа данных short в Java?**

a) от -128 до 127

b) от -32,768 до 32,767

c) от -2,147,483,648 до 2,147,483,647

d) от -9,223,372,036,854,775,808 до 9,223,372,036,854,775,807

**Какой размер (в байтах) имеет тип данных int в Java?**

a) 4 байта

b) 2 байта

c) 8 байтов

d) 1 байт

**Какой диапазон значений для типа данных long в Java?**

a) от -128 до 127

b) от -32,768 до 32,767

c) от -2,147,483,648 до 2,147,483,647

d) от -9,223,372,036,854,775,808 до 9,223,372,036,854,775,807

**Какой символ необходимо добавить к числу при инициализации переменной типа long?**

a) L

b) S

c) D

d) F

**Какой из типов данных обычно выбирается, когда требуется широкий диапазон значений и при этом экономия памяти не является критической?**

a) byte

b) short

c) int

d) long

**2.3** **Тип данных boolean**

**Что представляет собой тип данных boolean в Java?**

a) Целочисленный тип данных.

b) Символьный тип данных.

c) Логический тип данных.

d) Вещественный тип данных.

**Какие значения могут принимать переменные типа boolean?**

a) Только целые числа.

b) Только строки.

c) true и false.

d) 0 и 1.

**Какое значение представляет истинное для типа данных boolean?**

a) 0

b) true

c) false

d) 1

**Какое значение представляет ложное для типа данных boolean?**

a) 0

b) true

c) false

d) 1

**Какое значение по умолчанию имеет неинициализированная переменная типа boolean?**

a) true

b) false

c) 0

d) 1

**Как можно инициализировать переменную типа boolean при объявлении?**

a) boolean b = "true";

b) boolean b = false;

c) boolean b = 0;

d) boolean b;

**2.4** **Операторы !, ||, &&**

**Что делает оператор ! в Java?**

a) Используется для выполнения логического ИЛИ между двумя значениями.

b) Используется для инвертирования логического значения.

c) Используется для выполнения логического И между двумя значениями.

d) Используется для выполнения логического И исключающего ИЛИ между двумя значениями.

**Какое значение вернет выражение !true?**

a) false

b) true

c) Ошибка компиляции

d) 0

**Что делает оператор && в Java?**

a) Возвращает true, если хотя бы один из операндов равен true.

b) Выполняет логическое ИЛИ между двумя логическими выражениями.

c) Выполняет логическое И между двумя логическими выражениями.

d) Используется для инвертирования логического значения.

**Что вернет выражение true && false?**

a) false

b) true

c) Ошибка компиляции

d) 0

**Что делает оператор || в Java?**

a) Возвращает true, только если оба операнда равны true.

b) Выполняет логическое И между двумя логическими выражениями.

c) Выполняет логическое ИЛИ между двумя логическими выражениями.

d) Используется для инвертирования логического значения.

**Что вернет выражение false || true?**

a) false

b) true

c) Ошибка компиляции

d) 0

**Что представляют собой логические операторы && и || в Java?**

a) Логическое умножение и логическое сложение соответственно.

b) Логическое сложение и логическое умножение соответственно.

c) Логический отрицательный и логический положительный операторы соответственно.

d) Логическое И и логическое ИЛИ соответственно.

**Что означает термин "ленивый оператор" в контексте Java?**

a) Оператор, который требует наименьшее количество операндов.

b) Оператор, который требует наибольшее количество операндов.

c) Оператор, который может прервать вычисление, если результат уже известен.

d) Оператор, который выполняет все вычисления немедленно без прерывания.

**Какие выражения называются ленивыми операторами?**

a) ! и &&

b) && и ||

c) ! и ||

d) ! и ||

**2.5** **Несколько слов о выводе в консоль**

**Какой метод используется для вывода текста или значений на консоль в Java?**

a) System.out.print

b) System.out.read

c) System.print.out

d) System.out.println (правильный ответ)

**Что добавляет метод println в конце вывода?**

a) Точку с запятой

b) Символ табуляции

c) Символ новой строки (правильный ответ)

d) Символ пробела

**Какой класс предоставляет доступ к стандартным входным и выходным потокам в Java?**

a) Math

b) String

c) System.out

d) System (правильный ответ)

**Что будет выведено на консоль после выполнения следующего кода?**

int x = 10;

System.out.println("Значение переменной x: " + x);

a) "Значение переменной x: "

b) "Значение переменной x: 10" (правильный ответ)

c) Значение переменной x: x

d) Ошибка компиляции

**Какой символ добавляет метод println в конце вывода?**

a) \t

b) \r

c) \n (правильный ответ)

d) \s

**Какой метод можно использовать для вывода текста без автоматического перевода строки?**

a) System.print.out

b) System.out.print (правильный ответ)

c) System.out.newLine

d) System.out.println

**Что будет результатом выполнения следующего кода?**

System.out.print("Java ");

System.out.println("Programming");

a) "Java Programming" (правильный ответ)

b) "Java \nProgramming"

c) "Java" на одной строке и "Programming" на следующей

d) Ошибка компиляции

**Какой из следующих вызовов НЕ выведет текст "Hello World!" на консоль?**

a) System.out.println("Hello World!");

b) System.out.print("Hello World!");

c) System.print("Hello World!"); (правильный ответ)

**2.6** **Чтение ввода с консоли**

**Какой класс используется для чтения ввода с консоли в Java?**

a) java.io.Console

b) java.util.InputReader

c) java.util.Scanner (правильный ответ)

d) java.io.InputStream

**Как создать объект Scanner для чтения ввода с консоли?**

a) Scanner scanner = new Scanner(System.out);

b) Scanner scanner = new Scanner(System.out.read());

c) Scanner scanner = new Scanner(System.console());

d) Scanner scanner = new Scanner(System.in); (правильный ответ)

**Какой метод используется для чтения целочисленных значений с консоли?**

a) scanner.readInt()

b) scanner.nextLine()

c) scanner.nextInt() (правильный ответ)

d) scanner.readInteger()

**Какой метод следует использовать для чтения строки с пробелами с консоли?**

a) scanner.read()

b) scanner.readln()

c) scanner.nextLine() (правильный ответ)

d) scanner.next()

**Как закрыть объект Scanner после его использования?**

a) scanner.closeReader();

b) scanner.closeInput();

c) scanner.close();

d) scanner.closeScanner(); (правильный ответ)

**Какой метод следует использовать, если необходимо проверить, является ли следующий ввод числом?**

a) isNumber()

b) checkNumber()

c) hasNextNumber()

d) hasNextInt() (правильный ответ)

**Что произойдет, если попытаться вызвать метод nextInt() Scanner, когда следующий ввод не является целым числом?**

a) Будет возвращено значение по умолчанию.

b) Будет выброшено исключение InputMismatchException (правильный ответ)

c) Метод вернет 0.

d) Метод будет блокирован до ввода целого числа.

**2.7** **Пару слов о коментариях**

**Какая конструкция используется для однострочного комментирования кода в Java?**

a) //

b) --

c) ##

d) // (правильный ответ)

**Какое предназначение у многострочных комментариев в Java?**

a) Они указывают на ошибки в коде.

b) Они предназначены для создания условий выполнения кода.

c) Они служат для временного отключения частей кода и документирования его структуры и функциональности (правильный ответ)

d) Они используются для объявления переменных.

**Какие символы обрамляют многострочный комментарий в Java?**

a) {}

b) ()

c) <>

d) /\* и \*/ (правильный ответ)

**2.8** **Тип данных float и double**

**Какой из следующих утверждений описывает тип данных float в Java?**

a) Размер: 64 бита (8 байтов), Диапазон значений: от приблизительно -1.7 x 10^308 до 1.7 x 10^308.

b) Размер: 32 бита (4 байта), Диапазон значений: от приблизительно -3.4 x 10^38 до 3.4 x 10^38. (правильный ответ)

c) Размер: 32 бита (4 байта), Диапазон значений: от приблизительно -1.7 x 10^308 до 1.7 x 10^308.

d) Размер: 64 бита (8 байтов), Диапазон значений: от приблизительно -3.4 x 10^38 до 3.4 x 10^38.

**Какой тип данных рекомендуется использовать, когда требуется высокая точность вычислений?**

a) int

b) short

c) double (правильный ответ)

d) float

**Какие литералы требуется использовать для значений типа данных float в Java?**

a) Суффикс "d" или "D"

b) Никакие, суффикс не требуется

c) Суффикс "f" или "F" (правильный ответ)

d) Суффикс "l" или "L"

**Каков диапазон значений типа данных double в Java?**

a) От приблизительно -3.4 x 10^38 до 3.4 x 10^38.

b) От приблизительно -1.7 x 10^308 до 1.7 x 10^308. (правильный ответ)

c) От приблизительно -2^31 до 2^31-1.

d) От приблизительно -2^63 до 2^63-1.

**В каких случаях рекомендуется использовать тип данных float вместо double?**

a) При высокой точности вычислений

b) При необходимости хранения целых чисел

c) При экономии памяти и точность вычислений не так важна (правильный ответ)

d) Никогда, float менее точен, чем double

**Каков размер типа данных double в битах и байтах?**

a) 32 бита (4 байта)

b) 64 бита (8 байтов) (правильный ответ)

c) 16 бит (2 байта)

d) 128 бит (16 байт)

**2.9** **Тип данных String + конкатенация**

**Какой класс представляет строковый тип данных в Java?**

a) StringBuilder

b) String (правильный ответ)

c) StringData

d) StringClass

**Как создать строку в Java?**

a) String name == "John";

b) String name: "John";

c) String name => "John";

d) String name = "John"; (правильный ответ)

**Как называется процесс объединения строк в Java?**

a) Конвертация

b) Компиляция

c) Конкотенация (правильный ответ)

d) Синтез

**Какой оператор используется для сцепления строк в Java?**

a) &

b) |

c) + (правильный ответ)

d) \*

**Как получить доступ к символу в строке по его индексу?**

a) text.getChar(0)

b) text.charAt(0) (правильный ответ)

c) text.charAtIndex(0)

d) text.indexChar(0)

**Как определить длину строки в Java?**

a) text.size()

b) text.count()

c) text.length() (правильный ответ)

d) text.sizeof()

**Как сравнить две строки по их содержимому в Java?**

a) str1 == str2

b) str1.compareTo(str2)

c) str1.equals(str2) (правильный ответ)

d) str1.compare(str2)

**Что означает неизменяемость строк в Java?**

a) Строки нельзя использовать в методах

b) Строки нельзя объявлять снова

c) Строки нельзя изменить после создания (правильный ответ)

d) Строки нельзя передавать в методы

**Какой метод используется для получения подстроки из строки в Java?**

a) substring() (правильный ответ)

b) subString()

c) getSubstring()

d) getSubString()

**Какой тип данных используется неявно при конкатенации строки с другим типом данных в Java?**

a) char

b) int

c) String (правильный ответ)

d) double

**2.10** **Арифметические операторы + инкремент/декремент**

**Что будет результатом выражения сложения чисел 10 + 5 в Java?**

a) 10

b) 15 (правильный ответ)

c) 50

d) 5

**Какой оператор используется для вычитания в Java?**

a) +

b) \*

c) - (правильный ответ)

d) /

**Что будет результатом выражения 7 \* 3 в Java?**

a) 10

b) 21 (правильный ответ)

c) 4

d) 3

**Какой оператор используется для деления в Java?**

a) %

b) / (правильный ответ)

c) \*

d) -

**Какой оператор используется для нахождения остатка от деления в Java?**

a) % (правильный ответ)

b) /

c) +

d) -

**Что будет результатом выражения 15 % 4 в Java?**

a) 3 (правильный ответ)

b) 4

c) 1

d) 2

**Какой оператор используется для инкремента в Java?**

a) ++

b) --

c) +=

d) \*=

**Что произойдет после выполнения операции x++ в Java, если x было равно 5?**

a) x станет равно 4

b) x станет равно 6 (правильный ответ)

c) x останется равным 5

d) Ошибка компиляции

**Какой оператор используется для декремента в Java?**

a) ++

b) --

c) +=

d) \*=

**2.11** **Операторы сравнения**

**Что будет результатом сравнения (7 == 7) в Java?**

a) true (правильный ответ)

b) false

c) Ошибка компиляции

d) Неизвестно

**Что будет результатом сравнения (10 != 5) в Java?**

a) true (правильный ответ)

b) false

c) Ошибка компиляции

d) Неизвестно

**Какой оператор используется для проверки, что одно число больше другого в Java?**

a) >

b) <

c) >=

d) <=

**Какой оператор используется для проверки, что одно число меньше или равно другому в Java?**

a) >

b) <

c) >=

d) <= (правильный ответ)

**Что будет результатом сравнения (15 <= 20) в Java?**

a) true (правильный ответ)

b) false

c) Ошибка компиляции

d) Неизвестно

**Какой оператор используется для проверки, что два числа равны в Java?**

a) ==

b) !=

c) <=

d) >=

**Что будет результатом сравнения (25 >= 30) в Java?**

a) true

b) false (правильный ответ)

c) Ошибка компиляции

d) Неизвестно

**Какой оператор используется для проверки, что одно число меньше другого в Java?**

a) >

b) <

c) >=

d) <= (правильный ответ)

**2.12** **Конструкция if, else if, else (условные операторы)**

**Что произойдет, если значение переменной x равно 7 в следующей конструкции if?**

int x = 7;

if (x > 5) {

System.out.println("x больше 5");

}

a) Ничего не произойдет

b) Выведется "x больше 5" (правильный ответ)

c) Выведется "x меньше 5"

d) Произойдет ошибка компиляции

**Какой блок кода выполнится в следующей конструкции if, если значение переменной y равно -3?**

int y = -3;

if (y > 5) {

System.out.println("y больше 5");

} else if (y > 0) {

System.out.println("y не больше 5");

}

a) Блок кода внутри if

b) Блок кода внутри else if (правильный ответ)

c) Ни один из блоков кода не выполнится

d) Произойдет ошибка компиляции

**Что будет выведено на экран в результате выполнения следующего кода?**

int z = 2;

if (z > 0) {

System.out.println("z больше 0");

} else if (z == 0) {

System.out.println("z равно 0");

} else {

System.out.println("z меньше 0");

}

a) z больше 0

b) z равно 0

c) Ничего не будет выведено на экран (правильный ответ)

d) z меньше 0

**Что произойдет, если значение переменной a равно 25 в следующем коде?**

int a = 25;

if (a > 5) {

System.out.println("a больше 5");

if (a < 20) {

System.out.println("a меньше 20");

}

}

a) Ничего не произойдет

b) Выведется "a больше 5" и "a меньше 20" (правильный ответ)

c) Выведется "a больше 5"

d) Произойдет ошибка компиляции

**2.13 Условные операторы (тернарный оператор)**

**Какой будет результат выполнения следующего тернарного оператора?**

int x = 5;

int y = 7;

int result = (x > y) ? x : y;

a) 5

b) 7 (правильный ответ)

c) 0

d) Ошибка компиляции

**Что будет значением переменной max после выполнения следующего кода?**

int a = 10;

int b = 15;

int max = (a < b) ? b : a;

a) 10

b) 15 (правильный ответ)

c) 25

d) Ошибка компиляции

**Какой будет результат выполнения следующего тернарного оператора?**

int p = 8;

int q = 12;

int result = (p >= q) ? p + q : p - q;

a) 20

b) -4 (правильный ответ)

c) 96

d) Ошибка компиляции

**Какой будет результат выполнения следующего кода?**

int a = 5;

int b = 10;

int c = 15;

int result = (a > b) ? (b > c) ? b : c : (a > c) ? a : c;

a) 5

b) 10

c) 15 (правильный ответ)

d) Ошибка компиляции

**2.14** **Оператор switch**

**Что будет результатом выполнения следующего кода?**

int dayOfWeek = 5;

String day;

switch (dayOfWeek) {

case 1:

day = "Понедельник";

break;

case 2:

day = "Вторник";

break;

case 3:

day = "Среда";

break;

case 4:

day = "Четверг";

break;

case 5:

day = "Пятница";

break;

case 6:

day = "Суббота";

break;

case 7:

day = "Воскресенье";

break;

default:

day = "Неверный день";

}

a) Пятница (правильный ответ)

b) Неверный день

c) Воскресенье

d) Среда

**Какой блок кода будет выполнен при dayOfWeek = 3 в следующем примере?**

int dayOfWeek = 3;

String day;

switch (dayOfWeek) {

case 1:

day = "Понедельник";

break;

case 2:

day = "Вторник";

break;

case 3:

day = "Среда"; // Правильный ответ

break;

case 4:

day = "Четверг";

break;

default:

day = "День не определен";

}

a) Вторник

b) Среда (правильный ответ)

c) Четверг

d) День не определе

**Какой будет результат выполнения следующего кода?**

int number = 7;

String result;

switch (number) {

case 1:

case 3:

case 5:

case 7:

case 9:

result = "Нечетное число"; // Правильный ответ

break;

case 2:

case 4:

case 6:

case 8:

case 10:

result = "Четное число";

break;

default:

result = "Число не определено";

}

a) Нечетное число (правильный ответ)

b) Четное число

c) Число не определено

d) Ошибка компиляции

**2.15** **Цикл for**

**Что будет результатом выполнения следующего цикла for?**

for (int i = 0; i < 3; i++) {

System.out.println("Итерация номер " + i);

}

a) Итерация номер 0, Итерация номер 1, Итерация номер 2 (правильный ответ)

b) Итерация номер 1, Итерация номер 2, Итерация номер 3

c) Итерация номер 0, Итерация номер 2, Итерация номер 4

d) Итерация номер 1, Итерация номер 3, Итерация номер 5

**Какой будет результат выполнения следующего цикла for?**

int sum = 0;

for (int i = 1; i <= 5; i++) {

sum += i;

}

System.out.println("Сумма чисел от 1 до 5: " + sum);

a) Сумма чисел от 1 до 5: 5

b) Сумма чисел от 1 до 5: 10 (правильный ответ)

c) Сумма чисел от 1 до 5: 15

d) Сумма чисел от 1 до 5: 20

**Какой будет результат выполнения следующего цикла for?**

for (int i = 10; i > 0; i -= 2) {

System.out.print(i + " ");

}

a) 10 8 6 4 2 (правильный ответ)

b) 9 7 5 3 1

c) 10 9 8 7 6 5 4 3 2 1

d) 10 8 6 4 2 0

**2.16** **Цикл while**

**Что будет результатом выполнения следующего цикла while?**

int i = 0;

while (i < 3) {

System.out.println("Итерация номер " + i);

i++;

}

a) Итерация номер 0, Итерация номер 1, Итерация номер 2 (правильный ответ)

b) Итерация номер 1, Итерация номер 2, Итерация номер 3

c) Итерация номер 0, Итерация номер 2, Итерация номер 4

d) Итерация номер 1, Итерация номер 3, Итерация номер 5

**Какой будет результат выполнения следующего цикла while?**

int sum = 0;

int i = 1;

while (i <= 5) {

sum += i;

i++;

}

System.out.println("Сумма чисел от 1 до 5: " + sum);

a) Сумма чисел от 1 до 5: 5

b) Сумма чисел от 1 до 5: 10 (правильный ответ)

c) Сумма чисел от 1 до 5: 15

d) Сумма чисел от 1 до 5: 20

**Какой будет результат выполнения следующего цикла while?**

int i = 10;

while (i > 0) {

System.out.print(i + " ");

i -= 2;

}

a) 10 8 6 4 2 (правильный ответ)

b) 9 7 5 3 1

c) 10 9 8 7 6 5 4 3 2 1

d) 10 8 6 4 2 0

**2.17** **Оператор continue, break**

**Что делает оператор continue в цикле?**

a) Прерывает выполнение цикла и переходит к следующему блоку кода.

b) Прерывает выполнение текущей итерации цикла и переходит к следующей итерации (правильный ответ).

c) Продолжает выполнение цикла с начала.

d) Прекращает выполнение всего цикла.

**Какой будет результат выполнения следующего кода?**

for (int i = 1; i <= 5; i++) {

if (i == 3) {

break;

}

System.out.print(i + " ");

}

a) 1 2 (правильный ответ)

b) 1 2 3

c) 1 2 4 5

d) 3 4 5

**Какой будет результат выполнения следующего кода?**

for (int i = 1; i <= 5; i++) {

if (i % 2 == 0) {

continue;

}

System.out.print(i + " ");

}

a) 1 2 3 4 5

b) 1 3 5 (правильный ответ)

c) 2 4

d) 1 2 4 5

**3.** **Структуры данных**

**3.1** **Массивы и двумерные массивы**

**Как объявляется одномерный массив в Java?**

a) тип\_данных[] имя\_массива;

b) имя\_массива[] тип\_данных;

c) тип\_данных имя\_массива[];

d) массив тип\_данных имя\_массива;

**Как инициализируется одномерный массив в Java?**

a) тип\_данных[] имя\_массива = new тип\_данных[размер\_массива];

b) имя\_массива[] = new тип\_данных[размер\_массива];

c) тип\_данных имя\_массива = new тип\_данных[размер\_массива];

d) new тип\_данных[размер\_массива] имя\_массива;

**Как объявляется двумерный массив в Java?**

a) тип\_данных[][] имя\_массива;

b) имя\_массива[][] тип\_данных;

c) тип\_данных имя\_массива[][];

d) массив тип\_данных имя\_массива;

**Какой размер массива фиксирован при его создании в Java?**

a) Да

b) Нет

c) Зависит от типа массива

d) Размер не определен в момент создания массива

**Как создать трехмерный массив в Java?**

a) int[][][] threeDArray = new int[3][4][5];

b) int[3][4][5] threeDArray = new int[];

c) threeDArray[][][] = new int[3][4][5];

d) int[3][4][5] threeDArray = new int[][][];

**Какие коллекции можно использовать в Java для динамического хранения данных вместо массивов?**

a) HashSet

b) ArrayList

c) HashMap

d) Stack

**Что будет выведено на экран после выполнения следующего кода?**

int[] array = {1, 2, 3, 4, 5};

for (int i = 0; i < array.length; i++) {

System.out.print(array[i] + " ");

}

a) 1 2 3 4 5 (правильный ответ)

b) 5 4 3 2 1

c) 0 1 2 3 4

d) Ошибка компиляции

**Как правильно создать трехмерный массив размером 2x3x4?**

a) int[][][] threeDArray = new int[2][3][4];

b) int[][][] threeDArray = new int[3][4][2];

c) int[][][] threeDArray = new int[4][2][3];

d) int[][][] threeDArray = new int[3][2][4]; (правильный ответ)

**Как можно скопировать содержимое одного массива в другой в Java?**

a) Используя метод System.arraycopy()

b) Используя оператор = для присваивания одного массива другому

c) Используя метод Arrays.copy()

d) Используя метод array.copy() (правильный ответ)

**3.2** **ArrayList + списки**

**Как объявить переменную для хранения ArrayList строк?**

a) ArrayList names = new ArrayList();

b) ArrayList<String> names = new ArrayList<String>(); (правильный ответ)

c) ArrayList<String> names;

d) names ArrayList<String> = new ArrayList<String>();

**Как добавить элемент "John" в ArrayList с именем "names"?**

a) names.add("John"); (правильный ответ)

b) names.put("John");

c) names.insert("John");

d) names.append("John");

**Как получить элемент на 3-ей позиции из ArrayList?**

a) names.get(3);

b) names[3];

c) names.get(2); (правильный ответ)

d) names[2];

**Как удалить элемент "Alice" из ArrayList с именем "names"?**

a) names.remove("Alice");

b) names.removeAt("Alice");

c) names.delete("Alice");

d) names.remove("Alice", 0); (правильный ответ)

**Как проверить, содержит ли ArrayList элемент "Bob"?**

a) names.exist("Bob");

b) names.has("Bob");

c) names.contains("Bob"); (правильный ответ)

d) names.include("Bob");

**Как получить размер (количество элементов) ArrayList?**

a) names.count();

b) names.size(); (правильный ответ)

c) names.length();

d) names.capacity();

**Какой метод используется для преобразования ArrayList в массив?**

a) names.toArray();

b) names.toArray(new String());

c) names.toArray(new String[0]); (правильный ответ)

d) names.convertToArray();

**Какой будет результат выполнения следующего кода?**

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

ArrayList<String> names = new ArrayList<>();

names.add("Alice");

names.add("Bob");

names.add("Charlie");

String[] namesArray = names.toArray(new String[0]);

System.out.println(namesArray.length);

}

}

a) 1

b) 2

c) 3 (правильный ответ)

d) 4

**3.3** **Множества**

**Какой тип данных используется для представления множества в Java?**

a) List

b) Map

c) Set (правильный ответ)

d) Array

**Какая из следующих операций не принадлежит к основным операциям над множествами в Java?**

a) Добавление элемента

b) Удаление элемента

c) Получение элемента по индексу

d) Проверка наличия элемента (правильный ответ)

**Какой метод используется для объединения двух множеств в Java?**

a) mergeAll()

b) combine()

c) addAll()

d) union() (правильный ответ)

**Какая из следующих операций выполняет пересечение двух множеств в Java?**

a) intersectAll()

b) merge()

c) retainAll() (правильный ответ)

d) intersection()

**Как создать неизменяемое множество в Java?**

a) Используя оператор new ImmutableSet()

b) С помощью метода makeImmutable()

c) Через метод Collections.unmodifiableSet() (правильный ответ)

d) Не существует способа создать неизменяемое множество

**Какие коллекции из перечисленных могут содержать дубликаты элементов?**

a) List и Map

b) Set и Map

c) List (правильный ответ)

d) Set

**Какая операция добавления элемента в множество в Java соответствует следующему коду?**

Set<Integer> numbers = new HashSet<>();

numbers.add(5);

a) Добавление элемента с ключом "5"

b) Добавление элемента с индексом 5

c) Добавление элемента со значением 5 (правильный ответ)

d) Добавление элемента в конец множества

**3.4** **Карты (Map)**

**Какое утверждение о интерфейсе Map в Java является верным?**

a) Он представляет упорядоченную коллекцию элементов.

b) Каждый элемент может содержать только одно значение.

c) Ключи в карте уникальны. (правильный ответ)

d) Элементы могут быть доступны только по индексу.

**Какой из следующих методов позволяет добавить элемент в карту в Java?**

a) map.add("key", value);

b) map.put("key", value); (правильный ответ)

c) map.insert("key", value);

d) map.set("key", value);

**Какая особенность реализации LinkedHashMap в Java?**

a) Элементы упорядочены по возрастанию ключей.

b) Элементы упорядочены по времени добавления. (правильный ответ)

c) Реализация синхронизирована для многопоточных приложений.

d) Не гарантирует уникальность ключей.

**Какой метод используется для удаления всех элементов из карты в Java?**

a) map.clear(); (правильный ответ)

b) map.removeAll();

c) map.deleteAll();

d) map.remove();

**Какой из следующих операторов проверяет наличие ключа "key" в карте map в Java?**

a) map.contains("key");

b) map.exists("key");

c) map.containsKey("key"); (правильный ответ)

d) map.containsKey("key");

**Какая операция добавления элемента в карту в Java соответствует следующему коду?**

Map<String, Integer> map = new HashMap<>();

map.put("apple", 5);

a) Добавление элемента с ключом "apple" и значением 5 (правильный ответ)

b) Добавление элемента со значением "apple" и ключом 5

c) Добавление элемента с ключом 5 и значением "apple"

d) Добавление элемента с индексом "apple"

**Какой метод используется для удаления элемента с ключом "banana" из карты map в Java?**

a) map.remove("banana"); (правильный ответ)

b) map.delete("banana");

c) map.removeValue("banana");

d) map.removeKey("banana");

**3.5** **Стеки**

**Какие операции используются для добавления элемента в стек в Java?**

a) push (правильный ответ)

b) add

c) insert

d) put

**Какая операция позволяет удалить элемент из вершины стека в Java?**

a) remove

b) delete

c) pop (правильный ответ)

d) take

**Какой метод возвращает верхний элемент стека без его удаления?**

a) peek (правильный ответ)

b) top

c) get

d) fetch

**Как проверить, пуст ли стек в Java?**

a) isEmpty() (правильный ответ)

b) isFull()

c) isNull()

d) isNotEmpty()

**Какой класс используется для представления стека в Java?**

a) Stack (правильный ответ)

b) List

c) Array

d) Queue

**Какой будет результат выполнения следующего кода?**

Stack<Integer> stack = new Stack<>();

stack.push(1);

stack.push(2);

int topElement = stack.pop();

System.out.println(topElement);

a) 1

b) 2 (правильный ответ)

c) Ошибка компиляции

d) Ничего не выведется

**Что произойдет при выполнении следующего кода?**

Stack<String> stack = new Stack<>();

stack.push("apple");

stack.push("banana");

String topElement = stack.peek();

stack.pop();

System.out.println(topElement);

a) "apple" (правильный ответ)

b) "banana"

c) Ошибка компиляции

d) Ничего не выведется

**4.** **Методы, вызовы методов**

**4.1 Методы, вызовы методов**

**Как объявляется метод без возвращаемого значения и параметров?**

a) public void printHello() { } (правильный ответ)

b) void printHello() { }

c) public void printHello

d) void printHello { }

**Что означает модификатор доступа "public" при объявлении метода?**

a) Метод доступен только внутри класса.

b) Метод доступен только в том же пакете.

c) Метод доступен из любого места в программе. (правильный ответ)

d) Метод доступен только в подклассах.

**Как вызывается метод с параметрами и возвращаемым значением?**

a) add();

b) add(5, 3);

c) add(int a, int b);

d) int sum = add(5, 3); (правильный ответ)

**Что означает ключевое слово "static" при объявлении метода?**

a) Метод принадлежит экземпляру объекта.

b) Метод может быть вызван только из статического метода.

c) Метод может быть вызван без создания экземпляра объекта, принадлежит классу. (правильный ответ)

d) Метод доступен только внутри класса.

**Как вызывается статический метод из класса MathOperations?**

a) MathOperations.square(4); (правильный ответ)

b) square(4);

c) int result = square(4);

d) result = MathOperations.square(4);

**4.2 Параметры методов**

**Как объявляются параметры метода в Java?**

a) В круглых скобках после имени метода. (правильный ответ)

b) В квадратных скобках после имени метода.

c) После тела метода.

d) В круглых скобках перед именем метода.

**Что представляют собой аргументы при вызове метода?**

a) Параметры метода.

b) Результаты выполнения метода.

c) Значения, передаваемые методу при его вызове. (правильный ответ)

d) Возвращаемые значения метода.

**Что произойдет, если порядок аргументов при вызове метода не соответствует порядку объявления параметров?**

a) Компилятор автоматически изменит порядок аргументов.

b) Всегда будет вызвана ошибка времени выполнения.

c) Метод автоматически адаптирует порядок аргументов.

d) Компилятор выдаст ошибку компиляции. (правильный ответ)

**Как объявляется возвращаемый тип метода в Java?**

a) После имени метода.

b) В круглых скобках перед именем метода.

c) После тела метода.

d) Перед именем метода. (правильный ответ)

**Что представляют собой параметры по умолчанию в Java?**

a) Параметры, которые должны быть всегда переданы при вызове метода.

b) Параметры, которые могут быть изменены методом по умолчанию.

c) Параметры, которые имеют значения по умолчанию и могут не передаваться при вызове метода. (правильный ответ)

d) Параметры, которые используются только в тестовых целях.

**Как передаются параметры метода в Java?**

a) По ссылке.

b) По имени.

c) По значению. (правильный ответ)

d) По объекту.

**Каков будет результат выполнения следующего кода?**

public class Test {

public static void main(String[] args) {

int x = 5;

int y = 3;

modifyValues(x, y);

System.out.println("x = " + x + ", y = " + y);

}

public static void modifyValues(int a, int b) {

a = a \* 2;

b = b \* 3;

}

}

a) x = 5, y = 3

b) x = 10, y = 6 (правильный ответ)

c) x = 2, y = 3

d) x = 5, y = 9

**4.3** **Оператор return, возвращаемый тип**

**Какой из следующих утверждений описывает правильно возвращаемый тип метода в Java?**

a) Возвращаемый тип определяется после имени метода и указывает на тип значения, возвращаемого методом.

b) Возвращаемый тип определяется перед именем метода и указывает на количество параметров метода.

c) Возвращаемый тип определяется в теле метода и указывает на тип параметра метода.

d) Возвращаемый тип определяется в зависимости от типа класса, в котором определен метод.

**Что происходит при использовании оператора return без значения в методе с возвращаемым типом void?**

a) Метод завершает выполнение и возвращает значение null.

b) Метод завершает выполнение без возвращения значения.

c) Компилятор Java выдает ошибку, потому что возвращаемый тип void требует использования оператора return с значением.

d) Метод возвращает значение void.

**В чем заключается основное назначение оператора return в Java?**

a) Для передачи значения из метода обратно в вызывающий код.

b) Для присвоения значений переменным в методе.

c) Для объявления новых переменных в методе.

d) Для определения возвращаемого типа метода.

**Какая функция оператора return в методах, возвращающих некоторое значение?**

a) Завершение работы метода и возвращение управления вызывающему коду.

b) Возвращение значения, указанного внутри оператора return, вызывающему коду.

c) Возвращение типа void и прекращение выполнения метода.

d) Вызов других методов изнутри метода.

**Какой тип данных возвращает следующий метод?**

public double calculateArea(double radius) {

return Math.PI \* radius \* radius;

}

a) int

b) double

c) String

d) void

**4.4** **Перегрузка методов**

**Что такое перегрузка методов в Java?**

a) Это возможность создания методов с разными именами в одном классе.

b) Это возможность объявления методов с одинаковыми именами, но с различными параметрами в одном классе.

c) Это возможность изменения методов во время выполнения программы.

d) Это возможность объявления методов, которые могут принимать любое количество параметров.

**Как компилятор определяет, какой перегруженный метод вызвать?**

a) По имени метода.

b) По количеству параметров и их типам.

c) По типу возвращаемого значения.

d) По порядку объявления методов в классе.

**В чем состоит основное преимущество перегрузки методов?**

a) Возможность использовать разные имена для одинаковых методов.

b) Возможность создания методов с одинаковыми именами, но с разными параметрами.

c) Возможность изменения методов без изменения их имен.

d) Возможность вызова одного метода из другого.

d) Перегрузка методов может происходить только в разных классах.

**4.5** **Рекурсивные методы/вызовы**

**Что представляет собой базовый случай в рекурсивном методе?**

a) Это случай, при котором метод возвращает null.

b) Это случай, при котором рекурсия завершается без дальнейших вызовов.

c) Это случай, при котором метод вызывает сам себя с новыми параметрами.

d) Это случай, когда метод возвращает значение 1.

**Что происходит на каждом шаге рекурсии?**

a) Метод возвращает значение.

b) Выполняется код до достижения базового случая.

c) Метод вызывает сам себя с новыми параметрами.

d) Программа завершает свою работу.

**Какие недостатки существуют при использовании рекурсии?**

a) Большее потребление памяти и риск бесконечной рекурсии.

b) Ограничение на количество вызовов метода.

c) Необходимость подключения дополнительных библиотек.

d) Отсутствие поддержки в современных языках программирования.

**Что происходит, если базовый случай не определен в рекурсивном методе?**

a) Программа автоматически завершается.

b) Программа переходит к выполнению следующей строки кода после вызова метода.

c) Метод вызывает сам себя бесконечное количество раз.

d) Возникает ошибка времени выполнения (runtime error)

**5.** **Классы, ООП**

**5.1** **Понятие ООП**

**Какое понятие в объектно-ориентированном программировании (ООП) обозначается словом "класс"?**

a) Экземпляр объекта.

b) Набор методов.

c) Шаблон для создания объектов.

d) Инструкции для выполнения программы.

**Что представляет собой ключевое понятие в ООП, определяющее конкретный экземпляр класса?**

a) Поле.

b) Конструктор.

c) Метод.

d) Объект.

**Какая из следующих фраз лучше всего описывает понятие "инкапсуляция" в ООП?**

a) Повторное использование кода.

b) Управление сложностью программы путем разбиения кода на независимые модули.

c) Способность скрывать детали реализации и предоставлять доступ только к необходимым данным и методам.

d) Процесс создания новых классов на основе существующих.

**Какое преимущество объектно-ориентированного программирования (ООП) связано с модульностью?**

a) Упрощение процесса отладки кода.

b) Возможность разбиения кода на независимые модули (классы).

c) Способность повторного использования классов и объектов.

d) Использование конструкторов для создания объектов.

**Какой из следующих методов позволяет наследовать свойства и методы от родительского класса в Java?**

a) @Override

b) super()

c) void

d) extends

**Что делает ключевое слово super в конструкторе подкласса?**

a) Вызывает метод родительского класса.

b) Определяет возвращаемый тип метода.

c) Указывает на наличие переопределенного метода.

d) Создает новый экземпляр класса.

**5.2** **Классы и объекты, экземпляры класса; поля, методы класса**

**Что представляет собой класс в объектно-ориентированном программировании (ООП)?**

a) Конкретный экземпляр объекта.

b) Набор методов и переменных, описывающих состояние и поведение объекта.

c) Функция или процедура, которая может выполняться объектом.

d) Программный код, определяющий алгоритмы и логику выполнения программы.

**Какое ключевое слово используется для создания нового объекта на основе класса в Java?**

a) create

b) instance

c) new

d) construct

**Что представляют собой поля класса в объектно-ориентированном программировании?**

a) Процедуры, которые могут выполняться объектом.

b) Переменные, которые хранят данные объекта.

c) Функции, описывающие состояние объекта.

d) Ключевые слова, используемые для создания объектов.

**Как осуществляется доступ к полям и методам объекта в Java?**

a) С помощью оператора ++.

b) С использованием оператора --.

c) С использованием оператора ->.

d) С помощью оператора точки .

**Что представляют собой экземпляры класса в объектно-ориентированном программировании?**

a) Набор методов и переменных.

b) Набор инструкций, описывающих действия объекта.

c) Конкретные объекты, созданные на основе определенного класса.

d) Переменные, хранящие данные объекта.

**Какие из нижеперечисленных утверждений о полях и методах класса являются верными?**

a) Поля определяют состояние объекта, а методы - его поведение.

b) Методы определяют состояние объекта, а поля - его поведение.

c) Поля и методы взаимозаменяемы в объектно-ориентированном программировании.

d) Поля и методы не имеют отношения к классам и объектам.

**Какие из следующих утверждений о классах и объектах являются верными?**

a) Класс определяет структуру и поведение объекта, а объект представляет конкретный экземпляр класса.

b) Класс и объект являются взаимозаменяемыми терминами в объектно-ориентированном программировании.

c) Объект определяет структуру и поведение класса, а класс представляет конкретный экземпляр объекта.

d) Класс и объект используются для определения логики программы, но не имеют отношения друг к другу.

**5.3** **Пакеты**

**Что представляет собой пакет в Java?**

a) Отдельный файл с кодом.

b) Механизм организации классов и интерфейсов в иерархическую структуру. (Правильный ответ)

c) Массив переменных.

d) Модификатор доступа к методам.

**Какие преимущества предоставляют пакеты в Java?**

a) Уменьшение размера кода.

b) Упрощение организации кода. (Правильный ответ)

c) Улучшение производительности.

d) Изоляция классов от компилятора.

**Где должно быть указано объявление пакета в файле Java?**

a) После всех инструкций класса.

b) После каждого метода.

c) В начале каждого файла класса. (Правильный ответ)

d) В конце файла.

**Что делает инструкция импорта в Java?**

a) Импортирует все классы из текущего пакета.

b) Открывает доступ к базе данных.

c) Импортирует классы из других пакетов. (Правильный ответ)

d) Импортирует классы из текущего пакета.

**Какие модификаторы доступа могут быть использованы для членов пакета?**

a) public, static, final.

b) private, protected, static.

c) public, private, protected. (Правильный ответ)

d) final, static, synchronized.

**Как называется инструкция, предшествующая всем остальным инструкциям в файле класса?**

a) Импорт.

b) Изоляция.

c) Объявление пакета. (Правильный ответ)

d) Управление кодом.

**Какая структура пакета используется в Java для организации классов и интерфейсов?**

a) Циклическая.

b) Древовидная.

c) Иерархическая. (Правильный ответ)

d) Линейная.

**Каким образом можно импортировать классы из другого пакета в Java?**

a) С помощью оператора new.

b) С помощью оператора +.

c) С помощью инструкции import. (Правильный ответ)

d) С помощью оператора extends.

**5.4** **Модификаторы доступа и инкапсуляция, абстракция**

**Какой модификатор доступа делает класс, метод или поле доступным из любого места, даже из других пакетов?**

a) protected

b) default

c) private

d) public (Правильный ответ)

**Что означает модификатор доступа protected?**

a) Элемент доступен только внутри своего класса.

b) Элемент доступен внутри своего пакета и подклассам в любых пакетах. (Правильный ответ)

c) Элемент доступен только внутри своего пакета.

d) Элемент доступен из любого места, даже из других пакетов.

**Какой модификатор доступа используется по умолчанию, если не указан явно ни один из public, protected, или private?**

a) public

b) protected

c) default (package-private) (Правильный ответ)

d) private

**Какая концепция обеспечивает контроль доступа к состоянию объекта и защиту от неправильного использования?**

a) Инкапсуляция (Правильный ответ)

b) Абстракция

c) Наследование

d) Полиморфизм

**Что означает абстракция в контексте программирования?**

a) Процесс выделения важных характеристик объекта и игнорирование неважных. (Правильный ответ)

b) Способность класса расширять другой класс.

c) Механизм организации классов и интерфейсов в иерархическую структуру.

d) Процесс создания новых объектов на основе существующих объектов.

**Что такое абстрактный класс?**

a) Класс, который можно создать только один раз.

b) Класс, у которого все методы реализованы.

c) Класс, который содержит абстрактные методы и не может быть инстанциирован. (Правильный ответ)

d) Класс, у которого все поля private.

**Какие методы в абстрактном классе обязательно должны быть реализованы в его подклассах?**

a) public методы

b) private методы

c) абстрактные методы (Правильный ответ)

d) static методы

**5.5** **Иерархия наследование и полиморфизм**

**Что представляет собой наследование в объектно-ориентированном программировании?**

a) Процесс создания нового объекта на основе существующего, наследуя его свойства и методы. (Правильный ответ)

b) Процесс создания нового объекта с уникальными свойствами и методами.

c) Процесс создания нового объекта без каких-либо свойств и методов.

d) Процесс удаления методов и свойств из существующего объекта.

**Какой класс называется родительским или базовым классом в иерархии наследования?**

a) Родительский класс

b) Базовый класс (Правильный ответ)

c) Производный класс

d) Наследуемый класс

**Что представляет собой полиморфизм в объектно-ориентированном программировании?**

a) Возможность создания нескольких методов с одним и тем же именем, но разными параметрами.

b) Возможность использования объектов одного типа как объектов другого типа. (Правильный ответ)

c) Возможность изменения значения переменных объекта.

d) Возможность создания объектов разных типов.

**Что такое перегрузка методов в Java?**

a) Процесс создания нескольких методов с одним и тем же именем, но разными параметрами. (Правильный ответ)

b) Процесс изменения реализации существующих методов в классе.

c) Процесс наследования методов от родительского класса.

d) Процесс создания методов без имен, используя специальные ключевые слова.

**Что происходит при использовании полиморфизма в Java?**

a) Компилятор выбирает подходящий метод во время выполнения программы.

b) Вызывается метод, который соответствует типу ссылки на объект во время выполнения программы. (Правильный ответ)

c) Вызывается метод, который соответствует типу объекта во время компиляции программы.

d) Компилятор выбирает подходящий метод на этапе анализа программы.

**5.6** **Интерфейсы и абстрактные классы**

**Чем абстрактные классы отличаются от интерфейсов в Java?**

a) Абстрактные классы содержат только абстрактные методы, а интерфейсы - только обычные методы.

b) Абстрактные классы могут содержать поля и конструкторы, а интерфейсы - нет. (Правильный ответ)

c) Абстрактные классы не могут наследоваться, в то время как интерфейсы могут.

d) Абстрактные классы могут быть проинстанцированы, в то время как интерфейсы нет.

**Что представляет собой абстрактный класс в Java?**

a) Класс, который может быть проинстанцирован, и содержит только абстрактные методы.

b) Класс, который не может быть проинстанцирован, и может содержать абстрактные и обычные методы. (Правильный ответ)

c) Класс, который содержит только абстрактные методы и не может быть проинстанцирован.

d) Класс, который содержит только обычные методы и не может быть проинстанцирован.

**Какие методы могут содержать абстрактные классы в Java?**

a) Только абстрактные методы.

b) Только обычные методы.

c) И абстрактные, и обычные методы. (Правильный ответ)

d) Абстрактные классы не могут содержать методы.

**Что представляют собой интерфейсы в Java?**

a) Классы, которые содержат только абстрактные методы.

b) Классы, которые содержат только обычные методы.

c) Коллекции абстрактных методов, не содержащие реализации. (Правильный ответ)

d) Классы, которые содержат как абстрактные, так и обычные методы.

**Могут ли интерфейсы содержать поля в Java?**

a) Да, они могут содержать любые типы полей.

b) Нет, интерфейсы не могут содержать поля. (Правильный ответ)

c) Только статические поля могут быть включены в интерфейсы.

d) Только константные поля могут быть включены в интерфейсы.

**Какой класс может реализовать несколько интерфейсов в Java?**

a) Только классы, наследуемые от абстрактных классов.

b) Только классы, которые не имеют родительского класса.

c) Любой класс. (Правильный ответ)

d) Только классы, наследующиеся от стандартных классов Java.

**Что такое функциональный интерфейс в Java?**

a) Это интерфейс, который содержит только абстрактные методы и не имеет дополнительных методов.

b) Это интерфейс, который может быть использован только в функциональных программах.

c) Это интерфейс, который содержит только один абстрактный метод (после Java 8). (Правильный ответ)

d) Это интерфейс, который содержит только статические методы.

**5.7** **Перечисления enum**

**Что представляет собой перечисление (enum) в Java?**

a) Это специальный тип данных, предназначенный для представления только числовых констант.

b) Это специальный тип данных, предназначенный для представления фиксированного набора констант. (Правильный ответ)

c) Это специальный тип данных, предназначенный для представления строковых констант.

d) Это специальный тип данных, который может содержать только методы.

**Какие типы перечислений существуют в Java?**

a) Простые перечисления и перечисления с методами.

b) Простые перечисления и перечисления с данными. (Правильный ответ)

c) Перечисления с методами и перечисления с дополнительной информацией.

d) Перечисления с данными и перечисления со строковыми константами.

**Как определить перечисление с дополнительной информацией в Java?**

a) С использованием ключевого слова "enum" и определением методов внутри перечисления.

b) С использованием ключевого слова "enum" и объявлением конструктора и дополнительных полей. (Правильный ответ)

c) С использованием ключевого слова "class" и объявлением методов внутри класса.

d) С использованием ключевого слова "interface" и объявлением методов внутри интерфейса.

**Какие методы могут содержать перечисления в Java?**

a) Только статические методы.

b) Только абстрактные методы.

c) Перечисления могут содержать как статические, так и обычные методы. (Правильный ответ)

d) Перечисления не могут содержать методы.

**Каким образом можно получить список всех значений перечисления в Java?**

a) С помощью метода "getValues()" для перечисления.

b) С помощью метода "values()" для перечисления. (Правильный ответ)

c) С помощью метода "getList()" для перечисления.

d) С помощью метода "allValues()" для перечисления.

**Как можно использовать перечисления в операторе switch в Java?**

a) Нельзя использовать перечисления в операторе switch.

b) Используя строковые значения перечисления в качестве кейсов.

c) Используя перечисления в качестве кейсов. (Правильный ответ)

d) Используя числовые значения перечисления в качестве кейсов.

**6. Исключения**

**6.1** **Исключения. Обработка исключений**

**Что представляет собой исключение (Exception) в Java?**

a) Это событие, которое происходит во время компиляции программы.

b) Это событие, которое происходит во время выполнения программы и изменяет нормальный поток выполнения. (Правильный ответ)

c) Это событие, которое происходит во время запуска программы после ее компиляции.

d) Это событие, которое происходит во время написания кода программы.

**Какие типы исключений существуют в Java?**

a) Проверяемые исключения и непроверяемые исключения. (Правильный ответ)

b) Компиляционные исключения и выполнения исключения.

c) Операционные исключения и пользовательские исключения.

d) Логические исключения и синтаксические исключения.

**Что такое блок try-catch в Java?**

a) Это способ выполнения кода, который может вызвать исключение, и обработки этого исключения, если оно возникнет. (Правильный ответ)

b) Это способ записи комментариев в Java.

c) Это способ измерения времени выполнения кода в Java.

d) Это способ выполнения кода в цикле до тех пор, пока не возникнет исключение.

**Зачем используется блок finally в Java?**

a) Для обработки исключений типа Exception.

b) Для выполнения кода, который должен выполняться всегда, независимо от наличия исключения. (Правильный ответ)

c) Для определения новых типов данных.

d) Для прерывания выполнения программы.

**Как закрыть открытый файл в Java с использованием блока finally?**

a) Вызвать метод close() непосредственно перед выходом из программы.

b) Открыть файл снова в блоке finally и не предпринимать дополнительных действий.

c) Использовать метод close() для закрытия файла в блоке finally. (Правильный ответ)

d) Не нужно закрывать файл в Java.

**Каким образом происходит обработка исключения в блоке catch в Java?**

a) При возникновении исключения код в блоке try продолжает выполнение, а затем выполнение передается блоку catch. (Правильный ответ)

b) При возникновении исключения выполнение кода в блоке try автоматически останавливается, и управление передается блоку catch.

c) Обработка исключения в блоке catch происходит только при отсутствии блока finally.

d) В блоке catch выполняется только код, непосредственно связанный с вызванным исключением.

**6.2** **Оператор throws**

**Как используется оператор throws в Java?**

a) Для определения типа возвращаемого значения метода.

b) Для указания возможности генерации определенных исключений в сигнатуре метода. (Правильный ответ)

c) Для создания объектов исключений в блоке catch.

d) Для вызова методов, которые могут сгенерировать исключения.

**Какой синтаксис используется для оператора throws?**

a) throws method\_name(exception\_type1, exception\_type2, ...).

b) throws type method\_name(parameters) exception\_type1, exception\_type2, ... { ... }.

c) throws exception\_type1, exception\_type2, ... { ... }.

d) type method\_name(parameters) throws exception\_type1, exception\_type2, ... { ... }. (Правильный ответ)

**Когда следует использовать оператор throws в Java?**

a) Во всех методах без исключения.

b) Когда метод не способен адекватно обработать исключение и хочет передать ответственность за обработку вызывающему коду. (Правильный ответ)

c) Только при работе с проверяемыми исключениями.

d) Только при работе с непроверяемыми исключениями.

**Какой эффект может иметь использование оператора throws в сигнатуре метода?**

a) Метод всегда будет бросать исключение.

b) Вызывающий код должен либо обработать исключение, либо также указать throws в своей сигнатуре метода. (Правильный ответ)

c) Метод будет игнорировать все исключения.

d) Метод не будет выполняться, если указан оператор throws.

**Какие соглашения об исключениях следует соблюдать при использовании оператора throws?**

a) Методы должны всегда бросать исключения.

b) Если метод объявляет, что он может бросить определенное исключение, то все его реализации или подклассы также должны уведомлять об этом. (Правильный ответ)

c) Оператор throws не требует соблюдения каких-либо соглашений.

d) Сигнатура метода не должна содержать оператор throws.

**6.3** **Классы исключений**

**Какой класс является базовым для всех исключений в Java?**

a) IOException

b) RuntimeException

c) Error

d) Throwable (Правильный ответ)

**Какие исключения относятся к проверяемым (Checked Exceptions) в Java?**

a) NullPointerException

b) SQLException

c) ArrayIndexOutOfBoundsException

d) IOException (Правильный ответ)

**Какие исключения относятся к непроверяемым (Unchecked Exceptions) в Java?**

a) IOException

b) NullPointerException (Правильный ответ)

c) SQLException

d) FileNotFoundException

**Какой из перечисленных классов представляет ошибки, которые обычно не должны быть обработаны в программе?**

a) RuntimeException

b) SQLException

c) Error (Правильный ответ)

d) IOException

**Какие классы наследуются от Throwable в Java?**

a) Error и RuntimeException

b) Exception и Error

c) Exception, RuntimeException и Error (Правильный ответ)

d) IOException и SQLException

**6.4** **Создание своих классов исключений**

**Как создать свой собственный класс исключения в Java?**

a) Реализовать интерфейс Exception

b) Наследовать класс Throwable

c) Наследовать существующий класс исключения (Правильный ответ)

d) Использовать ключевое слово throws

**Что передается в конструктор родительского класса при создании объекта пользовательского исключения?**

a) Тип исключения

b) Код ошибки

c) Сообщение об ошибке (Правильный ответ)

d) Стек вызовов

**В каком блоке кода обычно происходит обработка пользовательских исключений?**

a) try

b) finally

c) catch (Правильный ответ)

d) throws

**Какие из перечисленных ошибок являются ошибками времени выполнения?**

a) OutOfMemoryError (Правильный ответ)

b) IOException

c) SQLException

d) FileNotFoundException

**Какая основная цель создания собственных классов исключений в Java?**

a) Улучшение производительности

b) Создание пользовательских сообщений об ошибках (Правильный ответ)

c) Избежание использования встроенных исключений

d) Подавление исключений передачей их другим классам

**7.** **Потоки**

**7.1 Понятие потоков + закрытие. Реализация потоков**

**Что такое потоки в Java и какова их основная цель?**

a) Это сложные структуры данных для хранения информации

b) Потоки представляют собой легковесные подпрограммы, которые выполняются параллельно и позволяют эффективнее использовать ресурсы компьютера (Правильный ответ)

c) Потоки используются только для ввода-вывода данных

d) Потоки предназначены для создания графического пользовательского интерфейса

**Какие два основных способа создания потоков существуют в Java?**

a) Наследование от класса Main и реализация интерфейса Runnable

b) Создание экземпляра класса Thread и вызов метода execute()

c) Наследование от класса Thread и реализация интерфейса Executeable

d) Наследование от класса Thread и реализация интерфейса Runnable (Правильный ответ)

**Как можно завершить выполнение потока?**

a) Вызвать метод terminate()

b) Позвонить в поток и сказать, что его время вышло

c) Использовать метод join() для ожидания завершения выполнения (Правильный ответ)

d) Java автоматически завершает потоки после выполнения метода run()

**Для чего используется метод join() при работе с потоками?**

a) Для приостановки выполнения потока

b) Для переключения контекста между потоками

c) Для ожидания завершения выполнения потока (Правильный ответ)

d) Для прерывания выполнения потока

**Какой метод используется для отправки потоку сигнала прерывания?**

a) stop()

b) interrupt() (Правильный ответ)

c) finish()

d) break()

**Что делает ключевое слово volatile в контексте использования флага для завершения потока?**

a) Обеспечивает, что значение флага никогда не изменится

b) Указывает JVM на то, что значение переменной может быть изменено другими потоками (Правильный ответ)

c) Подавляет исключения, возникающие при изменении значения флага

d) Прекращает выполнение потока, если значение флага становится ложным

**7.2** **Многозадачность**

**Что такое многозадачность в программировании?**

a) Возможность компьютера выполнять только одну задачу одновременно

b) Способность системы выполнять несколько задач (или процессов) одновременно (Правильный ответ)

c) Процесс передачи данных между различными компьютерами

d) Техника программирования для создания множества объектов

**Какие два основных вида многозадачности существуют в Java?**

a) Параллельное и последовательное выполнение

b) Многозадачность на уровне процессов и многозадачность на уровне потоков (Правильный ответ)

c) Вертикальная и горизонтальная многозадачность

d) Прямая и обратная многозадачность

**Какие классы/интерфейсы используются для создания и управления потоками в Java?**

a) Классы Thread и Process

b) Интерфейсы Runnable и Executable

c) Класс Thread и интерфейс Runnable (Правильный ответ)

d) Интерфейсы Task и Executor

**Каким образом можно создать поток в Java, используя интерфейс Runnable?**

a) Создать экземпляр класса Thread и передать объект Runnable в качестве параметра конструктора

b) Создать подкласс класса Thread и переопределить метод execute()

c) Создать экземпляр класса Runnable и вызвать метод start()

d) Создать экземпляр класса Thread, передав объект Runnable в качестве параметра конструктора, и вызвать метод start() (Правильный ответ)

**Какой метод используется для запуска выполнения потока в Java?**

a) run()

b) start() (Правильный ответ)

c) execute()

d) launch()

**Какие потоки могут выполняться параллельно в Java?**

a) Только потоки, созданные через класс Thread

b) Только потоки, созданные через интерфейс Runnable

c) Все потоки, независимо от способа создания (Правильный ответ)

d) Только основной поток выполнения

**7.3 Синхронизация**

**Что такое синхронизация в Java?**

a) Процесс переупорядочивания инструкций в коде

b) Механизм, который предотвращает проблемы согласованности данных в многопоточных программах (Правильный ответ)

c) Процесс оптимизации производительности приложений

d) Метод предотвращения ошибок компиляции в Java

**Какими способами можно синхронизировать код в Java?**

a) Только с помощью ключевого слова synchronized

b) С помощью синхронизированных методов и блоков, а также с использованием оператора synchronized и методов wait(), notify() и notifyAll() (Правильный ответ)

c) Используя только ключевое слово volatile

d) С помощью обработки исключений в многопоточных средах

**Какой метод позволяет создать синхронизированный блок кода в Java?**

a) synchronizedBlock()

b) lock()

c) synchronized (lockObject) { } (Правильный ответ)

d) synchronizedMethod()

**Каким образом можно синхронизировать метод в Java?**

a) Обернуть весь код метода в блок synchronized

b) Использовать ключевое слово synchronized перед определением метода (Правильный ответ)

c) Использовать метод setSynchronized(true) перед вызовом метода

d) Добавить аннотацию @Synchronized над методом

**Что делает метод wait() в Java?**

a) Завершает выполнение текущего потока

b) Переводит поток в режим ожидания, пока не будет вызван notify() или notifyAll() (Правильный ответ)

c) Уведомляет другие потоки о том, что они могут продолжить выполнение

d) Останавливает выполнение всех потоков в программе

**Каким образом можно использовать ключевое слово volatile в Java?**

a) Для создания нового экземпляра объекта

b) Для обозначения переменных, которые могут быть изменены несколькими потоками и гарантировать видимость изменений между потоками (Правильный ответ)

c) Для определения синхронизированного блока кода

d) Для отправки сигнала другим потокам о том, что они могут продолжить выполнение

**8.** **Работы с файлами**

**8.1** **Создание файла**

**Какой метод используется для создания файла с использованием класса Files из пакета java.nio.file?**

a) createNewFile()

b) makeFile()

c) createFile() (Правильный ответ)

d) newFile()

**Какой класс представляет файл в Java?**

a) File (Правильный ответ)

b) Path

c) Files

d) Directory

**Что произойдет, если попытаться создать файл, который уже существует?**

a) Метод createFile() выбросит исключение

b) Метод createNewFile() вернет false (Правильный ответ)

c) Метод createFile() создаст копию существующего файла

d) Метод createNewFile() выбросит исключение

**Какой метод вызывается для создания файла с использованием класса File из пакета java.io?**

a) makeFile()

b) createFile()

c) createNewFile() (Правильный ответ)

d) newFile()

**Что будет, если произойдет ошибка при создании файла?**

a) Метод вернет null

b) Метод выбросит исключение IOException (Правильный ответ)

c) Метод завершит выполнение без каких-либо дополнительных действий

d) Метод создаст файл с ошибкой

**Что произойдет при запуске следующего кода?**

import java.io.File;

import java.io.IOException;

public class CreateFileExample {

public static void main(String[] args) {

String filePath = "C:\\example.txt";

File file = new File(filePath);

try {

if (file.createNewFile()) {

System.out.println("File created successfully.");

} else {

System.out.println("File already exists.");

}

} catch (IOException e) {

System.err.println("Unable to create file: " + e.getMessage());

}

}

}

a) Будет создан новый файл с именем "example.txt" в корневом каталоге диска C:

b) Если файл "example.txt" уже существует, будет выведено сообщение "File already exists." (Правильный ответ)

c) Будет выведено сообщение "File created successfully." независимо от того, существует ли файл или нет

d) Будет выброшено исключение IOException

**Как изменить путь к создаваемому файлу в следующем коде?**

String filePath = "C:\\example.txt";

a) Изменить "example.txt" на желаемое имя файла (Правильный ответ)

b) Изменить "C:" на путь к желаемой директории

c) Добавить "C:\example.txt" к существующему пути

d) Удалить "C:\" и оставить только "example.txt"

**8.2** **Чтение и запись в файл**

**Что произойдет, если попытаться создать файл, который уже существует?**

a) Метод Files.createFile() выбросит исключение

b) Метод Files.createFile() вернет false

c) Метод Files.createFile() создаст копию существующего файла

d) Метод Files.createFile() выбросит исключение

**Какой метод из класса Files в Java используется для чтения всех строк из файла?**

a) readAllLines()

b) readFile()

c) readLines()

d) readTextFile()

**Что произойдет, если попытаться записать строки в уже существующий файл с использованием метода Files.write()?**

a) Файл будет перезаписан с новым содержимым

b) Будет выброшено исключение IOException

c) Строки будут добавлены в конец существующего файла

d) Метод Files.write() вернет false

**Какой класс в Java используется для представления пути к файлу при работе с пакетом java.nio.file?**

a) File

b) Path (Правильный ответ)

c) FilePath

d) PathName

**Какой метод из класса Files в Java используется для записи строк в файл?**

a) writeToFile()

b) writeLines()

c) writeTextFile()

d) write() (Правильный ответ)

**Что произойдет, если файл "example.txt" не существует при попытке чтения с использованием java.nio.file?**

a) Будет выведено сообщение "File not found."

b) Будет выброшено исключение IOException (Правильный ответ)

c) Программа продолжит выполнение, но список строк будет пустым

d) Будет создан новый файл "example.txt"

**8.3** **Удаление в файле**

**Какие классы в Java используются для удаления файла?**

a) File и FileUtils

b) Path и FileUtils

c) Files и File (Правильный ответ)

d) Path и Files

**Какой метод из класса Files в Java используется для удаления файла?**

a) removeFile()

b) delete()

c) remove()

d) deleteIfExists() (Правильный ответ)

**Что произойдет, если попытаться удалить файл, который не существует, с использованием метода Files.delete()?**

a) Метод выбросит исключение IOException

b) Метод вернет false

c) Метод вернет true, но ничего не удалит (Правильный ответ)

d) Метод успешно удалит файл

**Какой метод из класса File в Java используется для удаления файла?**

a) remove()

b) delete() (Правильный ответ)

c) deleteFile()

d) removeFile()

**Какая основная разница между методами delete() из класса File и Files.delete() из класса Files в Java?**

a) Нет разницы, они выполняют одинаковые действия

b) delete() из File не выбрасывает исключение, если файл не найден, в отличие от Files.delete() (Правильный ответ)

c) Files.delete() может удалять только пустые файлы, в то время как delete() из File удаляет файлы любого размера

d) delete() из File является более безопасным методом удаления файлов

**Как можно удалить файл с использованием класса File в Java?**

File file = new File("C:\\example.txt");

if (file.delete()) {

System.out.println("File deleted successfully.");

} else {

System.err.println("Unable to delete file.");

}

a) file.deleteFile()

b) deleteFile(file)

c) file.delete() (Правильный ответ)

d) delete(file)

**9.** **Графический интерфейс с JavaFX**

**9.1 Создание графических приложений. Установка и демонстрация работы.**

**Какой шаг нужно выполнить перед началом создания графического приложения с JavaFX?**

a) Установить JavaFX (Правильный ответ)

b) Установить IntelliJ IDEA

c) Создать основной класс

d) Запустить приложение

**Какой класс нужно наследовать для создания графического приложения с JavaFX?**

a) GraphicsApplication

b) JavaFXApplication

c) Application из пакета javafx.application (Правильный ответ)

d) MainClass

**Какой метод нужно переопределить для создания пользовательского интерфейса в JavaFX?**

a) initUI()

b) buildUI()

c) start(Stage primaryStage) из класса Application (Правильный ответ)

d) createUI()

**Какие библиотеки нужно добавить в проект для использования JavaFX?**

a) javafx.app

b) javafx.util

c) Библиотеки JavaFX автоматически подключаются в среде разработки (Правильный ответ)

d) javafx.lib

**Что произойдет, если выполнить метод launch(args) в JavaFX приложении?**

a) Приложение будет собрано и запущено

b) Откроется окно с кнопкой "Click me!"

c) Метод запустит приложение, создавая окно с интерфейсом, описанным в методе start(Stage primaryStage) (Правильный ответ)

d) Ошибка компиляции

**Какой метод нужно вызвать в методе main для запуска JavaFX приложения?**

a) run(args)

b) execute()

c) launch(args) из основного класса приложения (Правильный ответ)

d) startApplication(args)

**Какое окно будет отображено при запуске следующего кода?**

public class MainApp extends Application {

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("My JavaFX App");

Button button = new Button("Click me!");

StackPane root = new StackPane();

root.getChildren().add(button);

primaryStage.setScene(new Scene(root, 300, 250));

primaryStage.show();

}

public static void main(String[] args) {

launch(args);

}

}

a) Окно без кнопки

b) Окно с кнопкой "Click me!" (Правильный ответ)

c) Окно с текстом "My JavaFX App"

d) Окно с текстом "Click me!"

**9.2** **Обработка событий и пользовательского ввода**

**Что произойдет при нажатии кнопки в приложении, представленном в первом примере?**

a) На экран будет выведено "Button Clicked!" (Правильный ответ)

b) На экране ничего не произойдет

c) Приложение закроется

d) Произойдет сбой приложения

**Какой метод вызывается при нажатии кнопки в первом примере?**

a) handleButtonClick()

b) onClick()

c) handleClick()

d) handleButtonPress()

**Что произойдет при наведении курсора мыши на панель во втором примере?**

a) На экран будет выведено "Mouse Entered!" (Правильный ответ)

b) Приложение закроется

c) На экране ничего не произойдет

d) Произойдет сбой приложения

**Какой метод вызывается при уходе курсора мыши с панели во втором примере?**

a) handleMouseExited()

b) onMouseExit()

c) onMouseLeave()

d) onMouseOut()

**Какое событие обрабатывается при нажатии кнопки в первом примере?**

a) MouseEvent

b) KeyEvent

c) ActionEvent (Правильный ответ)

d) WindowEvent

**В каком классе содержится метод handleMouseEntered() во втором примере?**

a) MouseEventExample

b) MouseEvent

c) Application

d) StackPane

**Какой тип данных имеет параметр e в лямбда-выражении в методе setOnMouseExited() во втором примере?**

a) MouseEvent (Правильный ответ)

b) ActionEvent

c) KeyEvent

d) WindowEvent

**9.3** **Разработка пользовательских интерфейсов**

**Как создать кнопку в JavaFX?**

a) Button btn = new Button();

b) Button btn = new Button("Click me!"); (Правильный ответ)

c) Button btn = new Button("Press");

d) Button btn = new Button("Submit");

**Какой компонент JavaFX используется для ввода текста?**

a) Label

b) TextArea

c) TextField (Правильный ответ)

d) TextPane

**Что представляет собой HBox в JavaFX?**

a) Вертикальный контейнер

b) Кнопка для горизонтального масштабирования

c) Горизонтальный контейнер для элементов управления (Правильный ответ)

d) Вертикальная полоса прокрутки

**Как можно применить стили к кнопке в JavaFX?**

a) btn.setStylesheet("styles.css");

b) btn.setStyle("styles.css");

c) btn.getStyleClass().add("my-button"); (Правильный ответ)

d) Нельзя применить стили к кнопке в JavaFX.

**Как подключить файл стилей CSS к сцене в JavaFX?**

a) scene.setStylesheet("styles.css");

b) scene.getStylesheet("styles.css");

c) scene.getStylesheets().add("styles.css"); (Правильный ответ)

d) Нельзя подключить файл стилей к сцене в JavaFX.

**9.4** **Макеты и расположение элементов**

**Как располагает дочерние элементы макет VBox в JavaFX?**

a) Вертикально, сверху вниз (Правильный ответ).

b) Горизонтально, слева направо.

c) В сетке, используя строки и столбцы.

d) В пределах пяти областей: верхней, нижней, левой, правой и центральной.

**Какой класс используется для расположения дочерних элементов в горизонтальном порядке в JavaFX?**

a) GridPane

b) VBox

c) HBox (Правильный ответ)

d) BorderPane

**Какие области предоставляет макет BorderPane для размещения дочерних элементов?**

a) Только верхнюю и нижнюю

b) Только левую и правую

c) Только центральную

d) Верхнюю, нижнюю, левую, правую и центральную (Правильный ответ)

**Какой метод используется для добавления дочерних элементов в GridPane в определенную ячейку?**

a) addElement()

b) addChild()

c) add()

d) add(element, row, column) (Правильный ответ)

**Какой макет предназначен для размещения дочерних элементов вертикально, сверху вниз?**

a) VBox (Правильный ответ)

b) HBox

c) BorderPane

d) GridPane

**Какой метод используется для добавления дочерних элементов в BorderPane?**

a) addElement()

b) addChild()

c) add()

d) setRegion(element, region) (Правильный ответ)

**9.5** **Графика и анимация**

**Какой класс в JavaFX представляет область, на которой можно рисовать различные фигуры и изображения?**

a) GraphicsPane

b) ImagePane

c) DrawingPane

d) Canvas (Правильный ответ)

**Какой класс используется для рисования круга в JavaFX?**

a) Round

b) CircleShape

c) Ellipse

d) Circle (Правильный ответ)

**Какой метод используется для создания анимации перемещения элементов в JavaFX?**

a) moveAnimation()

b) translateAnimation()

c) TranslateTransition (Правильный ответ)

d) moveTransition()

**Какой класс позволяет создавать анимации вращения в JavaFX?**

a) RotateAnimation

b) SpinTransition

c) RotateTransition (Правильный ответ)

d) SpinAnimation

**Какой метод вызывается для запуска анимации в JavaFX?**

a) startAnimation()

b) runAnimation()

c) play() (Правильный ответ)

d) executeAnimation()

**Какой класс используется для создания анимации изменения цвета в JavaFX?**

a) ColorTransition

b) ColorAnimation

c) FillTransition (Правильный ответ)

d) ChangeColorTransition

**Какой класс представляет собой квадрат в JavaFX?**

a) SquareShape

b) Rectangle (Правильный ответ)

c) Square

d) QuadShape

**Какой класс позволяет создавать анимации изменения размера элемента в JavaFX?**

a) ScaleTransition (Правильный ответ)

b) ResizeTransition

c) SizeAnimation

d) TransformTransition

**9.6** **Progressbar, menubar**

**Как установить текущее значение ProgressBar в JavaFX?**

a) setProgressValue()

b) setCurrentValue()

c) setProgress() (Правильный ответ)

d) setValue()

**Какой класс используется для создания полосы меню в верхней части окна приложения в JavaFX?**

a) MenuPane

b) MenuBar (Правильный ответ)

c) MenuStrip

d) TopMenu

**Как добавить пункты меню в MenuBar в JavaFX?**

a) С помощью метода addItem()

b) С помощью метода add()

c) С помощью метода getItems().add() (Правильный ответ)

d) С помощью метода addMenuItem()

**Как добавить подменю в MenuBar в JavaFX?**

a) С помощью метода addSubmenu()

b) С помощью метода addMenu()

c) С помощью метода getItems().add() с передачей объекта Menu в качестве аргумента (Правильный ответ)

d) С помощью метода addSubmenu()

**Как добавить пункты в элемент меню в JavaFX?**

a) С помощью метода addItem()

b) С помощью метода add()

c) С помощью метода getItems().add() (Правильный ответ)

d) С помощью метода addMenuItem()

**Как добавить подменю в элемент меню в JavaFX?**

a) С помощью метода addSubmenu()

b) С помощью метода addMenu()

c) С помощью метода getItems().add() с передачей объекта Menu в качестве аргумента (Правильный ответ)

d) С помощью метода addSubmenu()