Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Косенок Сергей Михайлович

Должность: ректор

Дата подписания: 23.06.2025 14:53:08

Уникальный программный ключ: еза68f3eaa1e62674b54f4998099d3d6bfdcf836

Программирование на языке Java

Код, направление	27.03.04		
подготовки	Управление в технических системах		
Направленность	Инженерия автоматизированных, информационных и		
(профиль)	робототехнических систем		
Форма обучения	очная		
Кафедра-	Автоматики и компьютерных систем		
разработчик			
Выпускающая	Автоматики и компьютерных систем		
кафедра			

Тестовое задание для диагностического тестирования по дисциплине

Nº	Проверяемая компетенция	Задание	Тип сложности вопроса	
1	ПК-3.3	Каждый файл должен называться	низкий	
	ПК-6.2	* * * * * * * * * * * * * * * * * * *		
ПК-7.1		2. по имени класса в нём		
	ПК-7.2	3. по имени первой библиотеки в нём		
	ПК-8.2			
2	ПК-3.3	Что выведет этот код?	низкий	
	ПК-6.2	int a = 9;	пизкии	
	ПК-7.1	switch (a) {		
	ПК-7.2	<pre>case 0: System.out.print ("0");</pre>		
	ПК-8.2	case 5: System.out.print ("5"); break;		
		<pre>case 9: System.out.print ("9");</pre>		
		case 10: System.out.print ("10"); break;		
		<pre>default: System.out.print ("!");</pre>		
		}		
		1. 9		
		2. 10		
		3. Ошибка в коде		
		4. 910!		
		5. 910		
3	ПК-3.3	Какая аннотация используется при переопределении метода?	низкий	
	ПК-6.2	1. @Override		
	ПК-7.1	2. Override		
	ПК-7.2	3. @Redefine		
	ПК-8.2	4. Redefine		
4	ПК-3.3	Что общего у всех элементов массива?	низкий	
	ПК-6.2	1. Их названия		
	ПК-7.1	2. Их адрес в памяти		
	ПК-7.2	3. Их тип данных		
	ПК-8.2	4. Их размер		
5	ПК-3.3	По какому принципу работает стэк?	низкий	
	ПК-6.2	1. CY		
	ПК-7.1	2. LIFO		
	ПК-7.2	3. FIFO		
	ПК-8.2	4. FILO		
6	ПК-3.3	Какие числа будут выведены после выполнения цикла?	средний	
	ПК-6.2	for (int i = 10; i < 20; i += 2) {		
	ПК-7.1	if (i > 15) break;		
	ПК-7.2	if (i % 4 == 0) continue;		
	ПК-8.2	System.out.println (i);		
		}		
		1. 10		
		2. 12, 14		
	3. 11, 13, 15			
		4. 10, 12, 14 5. 14		
7	ПК-3.3		one Turi	
'	ПК-3.3 ПК-6.2	В каком состоянии не может пребывать поток? 1. TERMINATED	средний	
	ПК-6.2	2. RUNNABLE		
1	ПК-7.1	2. RUNNABLE 3. PAUSED		
1	ПК-7.2	4. NEW		
8	ПК-3.3	4. NEW Vector - реализация интерфейса:	средний	
	ПК-6.2	1. List	СРЕДПИИ	
	ПК-0.2	2. Map		
1	ПК-7.1	3. System		
1	ПК-7.2	4. Queue		
9	ПК-3.3	т. Queuc Какие режимы открытия файла существуют в Java?	средний	
	ПК-6.2	1. rws, rwd	Родини	
	ПК-7.1	2. r, rw, rws, rwd		
	ПК-7.2	3. r, w, s, r+		
	ПК-7.2	4. r, w, s, r+, s+, w+		
10	ПК-3.3	Почему нельзя наследоваться от класса string?	средний	
	ПК-6.2	1. Он не является классом	-F-7	
	ПК-7.1	2. Он не имеет тела		
		1	1	

, ,			
ł	ПК-7.2	3. Он объявлен как final	
	ПК-8.2	4. От класса string можно наследоваться	
11	ПК-3.3	Где правильно создана простая переменная?	средний
	ПК-6.2	1. int[] a;	
	ПК-7.1	 byte x = 100000; char str = 'ab'; 	
	ПК-7.2		
	ПК-8.2	4. bool isDone = true;	
		5. float $x = 23.3f$;	
12	ПК-3.3	Где правильно создан массив?	средний
	ПК-6.2	1. $int[]$ a = new int $\{1, 2, 3, 4, 5\}$;	
	ПК-7.1	2. int $a = new int[] \{1, 2, 3, 4, 5\};$	
	ПК-7.2	3. $int[] a = int[] \{1, 2, 3, 4, 5\};$	
	ПК-8.2	4. int[] a = new int[] {1, 2, 3, 4, 5};	
10	HI4 2 2	5. int a[] = 1, 2, 3, 4, 5;	.,
13	ПК-3.3	Какой метод переводит поток в режим ожидания?	средний
	ПК-6.2	1. pause()	
	ПК-7.1	2. wait()	
	ПК-7.2	3. stop()	
1.4	ПК-8.2	4. break()	U
14	ПК-3.3	Каким методом можно перевернуть строку?	средний
	ПК-6.2	1. string.reverse();	
	ПК-7.1	2. stringBuilder.reverse();	
	ПК-7.2	3. string.return();	
1.5	ПК-8.2	4. stringBuilder. return();	
15	ПК-3.3	В чем здесь ошибка?	средний
	ПК-6.2	int a, b;	
	ПК-7.1	System.out.print("Введите первое число: ");	
	ПК-7.2	<pre>Scanner num = new Scanner(System.in); a = num.nextFloat();</pre>	
	ПК-8.2	a = num.nextrioat(); 1. Ошибок нет	
		2. Вместо System.in надо использовать System.out	
		2. вместо System. In надо использовать System.out 3. int a, b - необходимо записывать по отдельности	
		4. Вместо nextFloat надо использовать nextInt	
16	ПК-3.3	Как называется процесс сохранения состояния объекта в	высокий
10	ПК-5.3	последовательность байт?	BDICOKHH
		Ответ:	
,)	11K'_7 1		
	ПК-7.1 ПК-7.2	O'IBC1	
	ПК-7.2	O'Be'l.	
17	ПК-7.2 ПК-8.2		высокий
17	ПК-7.2 ПК-8.2 ПК-3.3	Что выведет данная функция?	высокий
17	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2	Что выведет данная функция? public void test() {	высокий
17	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1	<pre>Что выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717);</pre>	высокий
17	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2	Что выведет данная функция? public void test() {	высокий
17	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2	<pre>Что выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717);</pre>	высокий
17	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2	<pre>Что выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2);</pre>	высокий
17	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2	<pre>Что выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); }</pre>	высокий
	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2	<pre>Uto выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } Oтвет:</pre>	
	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2	Что выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } Ответ: Какой результат выведет вызов метода foo()?	
	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2	<pre>Uro выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } Otbet: Какой результат выведет вызов метода foo()? void foo() {</pre>	
	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2	<pre>Uro выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } Otbet: Kakoй результат выведет вызов метода foo()? void foo() { String m = "Hello"; System.out.print(m); bar(m);</pre>	
	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2	<pre>Uro выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } Oтвет: Какой результат выведет вызов метода foo()? void foo() { String m = "Hello"; System.out.print(m);</pre>	
	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2	<pre>UTO выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } OTBET: Kакой результат выведет вызов метода foo()? void foo() { String m = "Hello"; System.out.print(m); bar(m); System.out.print(m); }</pre>	
	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2	<pre>UTO выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } OTBET: Kакой результат выведет вызов метода foo()? void foo() { String m = "Hello"; System.out.print(m); bar(m); System.out.print(m); } void bar(String m) {</pre>	
	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2	<pre>UTO выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } OTBET: Kакой результат выведет вызов метода foo()? void foo() { String m = "Hello"; System.out.print(m); bar(m); System.out.print(m); }</pre>	
	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2	<pre>UTO выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } OTBET: Kакой результат выведет вызов метода foo()? void foo() { String m = "Hello"; System.out.print(m); bar(m); System.out.print(m); } void bar(String m) { m += " World!"; }</pre>	
18	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2	<pre>UTO выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } OTBET: Kакой результат выведет вызов метода foo()? void foo() { String m = "Hello"; System.out.print(m); bar(m); System.out.print(m); } void bar(String m) { m += " World!"; } OTBET:</pre>	высокий
	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2	<pre>UTO выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } OTBET: Kакой результат выведет вызов метода foo()? void foo() { String m = "Hello"; System.out.print(m); bar(m); System.out.print(m); } void bar(String m) { m += " World!"; } OTBET: Вставьте пропущенное слово:</pre>	
18	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2	Что выведет данная функция?public void test() {Integer i1 = Integer.valueOf(717);Integer i2 = Integer.valueOf(717);System.out.println(i1 == i2);}OTBET:Какой результат выведет вызов метода foo()?void foo() {String m = "Hello";System.out.print(m);bar(m);System.out.print(m);}void bar(String m) {m += " World!";}Ответ:Вставьте пропущенное слово:Исключения, порожденные от RuntimeException, являются	высокий
18	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-7.2 ПК-8.2	Что выведет данная функция?public void test() {Integer i1 = Integer.valueOf(717);Integer i2 = Integer.valueOf(717);System.out.println(i1 == i2);}OTBET:Kakoй результат выведет вызов метода foo()?void foo() {String m = "Hello";System.out.print(m);bar(m);System.out.print(m);}void bar(String m) {m += "World!";}Ответ:Вставьте пропущенное слово:Исключения, порожденные от RuntimeException, являются икомпилятор не требует обязательной их обработки.	высокий
18	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-7.2 ПК-8.2	Что выведет данная функция?public void test() {Integer i1 = Integer.valueOf(717);Integer i2 = Integer.valueOf(717);System.out.println(i1 == i2);}OTBET:Какой результат выведет вызов метода foo()?void foo() {String m = "Hello";System.out.print(m);bar(m);System.out.print(m);}void bar(String m) {m += " World!";}Ответ:Вставьте пропущенное слово:Исключения, порожденные от RuntimeException, являются	высокий
19	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-8.2	Что выведет данная функция?public void test() {Integer i1 = Integer.valueOf(717);Integer i2 = Integer.valueOf(717);System.out.println(i1 == i2);}OTBET:Kaкой результат выведет вызов метода foo()?void foo() {String m = "Hello";System.out.print(m);bar(m);System.out.print(m);}void bar(String m) {m += " World!";}Ответ:Вставьте пропущенное слово:Исключения, порожденные от RuntimeException, являются икомпилятор не требует обязательной их обработки.Ответ:	высокий
18	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-7.1 ПК-7.2 ПК-7.1 ПК-7.2 ПК-7.1	Что выведет данная функция?public void test() {Integer i1 = Integer.valueOf(717);Integer i2 = Integer.valueOf(717);System.out.println(i1 == i2);}OTBET:KAKOЙ РЕЗУЛЬТАТ ВЫВЕДЕТ ВЫЗОВ МЕТОДА foo()?void foo() {String m = "Hello";System.out.print(m);bar(m);System.out.print(m);}void bar(String m) {m += "World!";}ОТВЕТ:Вставьте пропущенное слово:Исключения, порожденные от RuntimeException, являются и компилятор не требует обязательной их обработки.ОТВЕТ:Что выведет этот код:	высокий
19	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-7.1 ПК-7.2 ПК-7.1 ПК-7.2 ПК-7.3 ПК-6.2	Что выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); } OTBET: Kakoй pesynьтат выведет вызов метода foo()? void foo() { String m = "Hello"; System.out.print(m); bar(m); System.out.print(m); } void bar(String m) { m += " World!"; } OTBET: BCTABЬТЕ пропущенное слово: Исключения, порожденные от RuntimeException, являются и компилятор не требует обязательной их обработки. ОТВЕТ: ЧТО выведет этот код: int a = 9;	высокий
19	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1	Что выведет данная функция?public void test() {Integer i1 = Integer.valueOf(717);Integer i2 = Integer.valueOf(717);System.out.println(i1 == i2);}OTBET:	высокий
19	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-8.2	Что выведет данная функция? public void test() { Integer i1 = Integer.valueOf(717); Integer i2 = Integer.valueOf(717); System.out.println(i1 == i2); Corber: Kakoй результат выведет вызов метода foo()? void foo() { String m = "Hello"; System.out.print(m); bar(m); System.out.print(m); } void bar(String m) { m += " World!"; } Other: Bcтавьте пропушенное слово: Исключения, порожденные от RuntimeException, являются и компилятор не требует обязательной их обработки. Ответ: Что выведет этот код: int a = 9; boolean isDone = false; if (a % 3 != 0 !isDone)	высокий
19	ПК-7.2 ПК-8.2 ПК-3.3 ПК-6.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1 ПК-7.2 ПК-7.1 ПК-7.2 ПК-8.2 ПК-7.1	Что выведет данная функция?public void test() {Integer i1 = Integer.valueOf(717);Integer i2 = Integer.valueOf(717);System.out.println(i1 == i2);}OTBET:	высокий