Взірець завдань для вступу за напрямом Java

Version 2.3

For

***SoftServe IT Academy***

*July, 2019*

Contents

[1 МЕТА 4](#_Toc383168216)

[2 Приклад завдання для вступу 5](#_Toc383168217)

[3 Література 10](#_Toc383168218)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
|  /05/2012 | 0.1 | Initial version | Yaroslav Harasym |
|  /05/2012 | 0.1 | Initial version | Lesya Klakovych |
| 25/05/2012 | 0.2 | Revision  | Mariya Mudra |
|  /03/2014 | 2.0 | New approach – entry test is a language based | Yaroslav Harasym |
|  /03/2014 | 2.0 | Initial version | Lesya Klakovych |
| 07/2018 | 2.1 | Corrected time for tests | Lesya Klakovych |
| 11/2018 | 2.2 | Deleting time limits | Lesya Klakovych |
| 12/2019 | 2.3 | Test samples updated | Yaroslav Harasym |
|  |  |  |  |

# МЕТА

Цей документ створений, для того, щоб зробити процес вступу на навчання у SoftServe IT Academy за напрямом Java простішим та зрозумілішим для кандидатів. Тут описані вимоги до рівня знань кандидатів, подані приклади завдань для тестування, рекомендовано онлайн-тест для визначення свого рівня та необхідну для навчання література.

# Приклад завдання для вступу

**Загальні положення**

Вступний тест складається з сорока питань які охоплюють базові розділи Java Core, основи функціонування мереж та баз даних.

1. Вступне випробування представляє собою набір тестових запитань закритого типу, які формуються випадковим чином із загальної бази. Присутні різні способи отримання відповідей. Наприклад, відповідь потрібно вносити у текстове поле у вигляді цілого числа, або ж вибирати варіант із випадаючого списку. Переважно, слід вибрати один або більше відповідей із запропонованих. Присутні також питання, в яких потрібно будувати частину коду перетягуючи мишкою фрагменти операторів із заданих варіантів. Результати тесту перевіряються автоматично і відомі негайно після проходження.
2. Використовувати середовище для розробника не можна. З цією метою переважна більшіть питань містить фрагмент коду у вигляді малюнка. Відповідно, час розрахований так, щоб його було достатньо для осмислення питання та формулювання відповіді, однак замало, щоб перенабирати і перевіряти код у середовищі розробника.
3. Мова, що використовується для вступного тестування. Всі пропоновані завдання сформульовані англійської мовою. Для коментування у деяких питаннях можна використовувати зручну Вам мову.

**Зразки завдань**

Зразок тестових запитань по Java:

**1**. **Which is a valid keyword in Java**?

1. interface
2. String
3. float
4. unsigned

**2**. **Which will legally declare, construct, and initialize an array**?

1. int [] myList = {"1", "2", "3"};
2. int [] myList = (5, 8, 2);
3. int myList [] [] = {4,9,7,0};
4. int myList [] = {4, 3, 7};

**3**. **Which three piece of codes are equivalent to line 2**?

**public** **interface** Foo {

 int k = 4; /\* Line 2 \*/

}

1. final int k = 4;
2. public int k = 4;
3. static int k = 4;
4. abstract int k = 4;
5. volatile int k = 4;
6. protected int k = 4;
7. 1, 2 and 3
8. 2, 3 and 4
9. 3, 4 and 5
10. 4, 5 and 6

**4**. **Which are valid declarations of a char**?

* 1. char c1 = 064770;
	2. char c2 = 'face';
	3. char c3 = 0xbeef;
	4. char c4 = \u0022;
	5. char c5 = '\iface';
	6. char c6 = '\uface';
1. 1, 2, 4
2. 1, 3, 6
3. 3, 5
4. 5 only

**5**. **What will be the output of the program**?

**try** {

 int x = 0;

 int y = 5 / x;

} catch (Exception e) {

 System.out.println("Exception");

} catch (ArithmeticException ae) {

 System.out.println("Arithmetic Exception");

}

System.out.println("finished");

1. finished
2. Exception
3. Compilation fails
4. Arithmetic Exception

**6. Consider classes Customer and Rental, which represent customer and information about rent.**

**class** Customer {

    private String name;

    // Other fields, constructors, get, set, etc.

    //

    public Customer(String name) {

        this.name=name;

    }

    private double amountFor(Rental rental) {

        double result=rental.getDays()\*rental.getBasePrice();

        if (rental.getKind()==1) {

            result=result\*1.5;

        }

        if (rental.getKind()==2) {

            result=result\*2;

        }

        if (rental.getKind()==3) {

            result=result\*2.5;

        }

        if (rental.getDays()>7) {

            result=result\*3;

        }

        // Another code.

        return result

    }

    // Other methods.

}

**class** Rental {

    private int kind;

    private int days;

    // Other fields, constructors, get, set, etc.

    //

    public Rental(int kind, int days) {

        this.kind=kind;

        this.days=days;

    }

    public double getBasePrice() {

        // Calculation of Price.

        // . . .

    }

    // Other methods.

}

Move method amountFor(…) into class Rental.

# Література

* Herbert Schildt. Java: The Complete Reference. New-York, Chicago, etc. 2018
* Кей С. Хорстманн, Гари Корнелл Java. Библиотека профессионала. Т. 1, 2. М.: "Вильямс", 2015.
* Брюс Эккель. Философия Java. 4-е изд. - СПб.: Питер, 2009. - 640 с.
* Joshua Bloch. Effective Java. Second Edition. – Upper Saddle River, NJ, Boston, Indianapolis, San Francisco, New York, Toronto, Montreal, London, Munich, Paris, Madrid, Capetown, Sydney, Tokyo, Singapore, Mexico City: Addison-Wesley, 2008.
* Yakov Fain. Java Programming. 24-Hour Trainer. – Indianapolis: Wiley Publishing, 2011.
* Анил Хемраджани. Гибкая разработка приложений на Java. – М.: Вильямс, 2008. – 352 с.