

Взірець завдань для вступу за напрямом Java

Version 2.0

For

SoftServe IT Academy

March, 2014

Contents

1	МЕТА.....	4
2	ПРИКЛАД ЗАВДАННЯ ДЛЯ ВСТУПУ	5
3	ЛІТЕРАТУРА	10

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	Newapproach – entry test is a language based	Yaroslav Harasym
/03/2014	2.0	Initial version	Lesya Klakovych

1 META

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.

2 Приклад завдання для вступу

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Вступний тест складається з двох частин:

1. Перша частина – набір тестових запитань закритого типу (слід вибрати один або більше і варіантів із запропонованих). Результати цього тесту перевіряються автоматично і відомі негайно після проходження. У тесті 10 запитань. Тривалість тесту 30 хвилин. Ця частина покриває теми по JavaCore;
2. До другої частини допускаються кандидати, що пройшли першу частину успішно. Ця частинка передбачає відповідь на запитання відкритого типу (необхідно вписати код у відповідне поле, прокоментувати, тощо). Тривалість другої частини – 60 хвилин. Використовувати середовище для розробника не можна. Ця частина тесту перевіряються вручну і її результати доступні після перевірки.
3. Мова, що використовується для вступного тестування. Всі пропонувані завдання будуть сформульовані англійською мовою. У другій частині передбачається Ваша словесна відповідь разом з кодом програми. Для коментування у відкритих питаннях можна використовувати зручну Вам мову.

ЗРАЗКИ ЗАВДАНЬ

Зразок тесту з першої частини:

1. Which is a valid keyword in java?

- a) interface
- b) string
- c) Float
- d) unsigned

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

- a) `int [] myList = {"1", "2", "3"};`
- b) `int [] myList = (5, 8, 2);`
- c) `int myList [] [] = {4, 9, 7, 0};`
- d) `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;`

- a) 1, 2 and 3
- b) 2, 3 and 4
- c) 3, 4 and 5
- d) 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';`

- a) 1, 2, 4
- b) 1, 3, 6
- c) 3, 5
- d) 5 only

5. What will be the output of the program?

```
try{  
    int x = 0;  
    int y = 5 / x;  
}
```

SoftServe

Public

```
} catch (Exception e) {  
    System.out.println("Exception");  
} catch (ArithmeticException ae) {  
    System.out.println("Arithmetic Exception");  
}  
System.out.println("finished");
```

- a) finished
- b) Exception
- c) Compilation fails
- d) Arithmetic Exception

Зразок завдання для другої частини:

TASK 1. 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;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name=name;
    }
    private double amountFor(Rental rental) {
        double result;
        // Another code.
        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;
        }
        // Other calculation.
        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 int getKind() {
        return kind;
    }
}
```



```

    public void setKind(int kind) {
        this.kind=kind;
    }
    public int getDays() {
        return days;
    }
    public void setDays(int days) {
        this.days=days;
    }
    public double getBasePrice() {
        // Calculation of Price.
        // . . .
    }
    // Other methods.
}

```

Move method `amountFor(...)` into class `Rental`. Give explanation of this moving. Propose the solution for improving code quality.

TASK 2. Create classes, which describe employees with hourly wage and fixed payment. Give your suggestions about relations between classes. Implement method for calculating the average monthly salary. For employees with hourly wage use next formula: “average monthly salary= 20.8*8* hourly rate”, for employees with fixed payment – “average monthly salary= fixed monthly payment”. Write well commented code for solving next problems

- Sort the collection of employees in descending order by the average monthly salary. In the case of equal salary – by the name. Write ID, name and monthly salary for all employees from collection.
- Write information about first five employees from collection (problem a).
- Write ID of three last employees from collection (problem b).
- Write code for reading and writing collection of these objects from (into) file.
- Write code for handling the incorrect format of incoming file.

3 Література

- ✓ Брюс Эккель. Философия Java. 4-е изд. - СПб.: Питер, 2009. - 640 с.
- ✓ Кей С. Хорстманн, Гарри Корнелл Java 2, Т. 1, 2. М.: "Вильямс", 2007.
- ✓ Yakov Fain. Java Programming. 24-Hour Trainer. – Indianapolis: Wiley Publishing, 2011.
- ✓ 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.
- ✓ Анил Хемраджани. Гибкая разработка приложений на Java. – М.: Вильямс, 2008. – 352 с.
- ✓ Ильдар Хабибуллин. Самоучитель Java. – СПб.: БХВ-Петербург, 2001. – 464 с.