

练习 时间: 20min

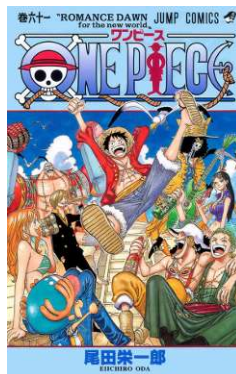
- 模拟简单计算器，可以运算 $+$ ， $-$ ， $*$ ， $/$ ， $\%$ 。
 - 接收三个参数，一个整数，一个运算符，另一个整数。
 - 计算出运算结果并输出。
 - 无法运算时，返回null。
- 使用Scanner类接收输入，
- 代码实现，效果如图所示：

```
3
/
4
3/4=0
```

```
4324
%
4
4324%4=0
```

练习 时间: 20min

- 初始化如下HashMap:
 - **key**: String类, **value**: ArrayList<String>类
 - 添加以下键值对:
 - “One Piece” – [“Monkey D. Luffy”, “Roronoa Zoro”, “Nami”, “Usopp”, “Vinsmoke Sanji”, “Tony Tony Chopper”, “Nico Robin”...]
 - “Naruto” – [“Naruto Uzumaki”, “Sasuke Uchiha”, “Sakura Haruno”...]
 - “Attack on Titan” – [“Eren Jaeger”, “Mikasa Ackerman”, “Armin Arlelt”...]
- 对HashMap进行遍历, 将每一个value的size输出出来



练习 时间：30min

- 做一个用户登录系统
- 要求实现功能：
 - 用户注册功能：
 - 选择合适的容器存储用户注册时输入的`userName`和`password`
 - 用户登录功能：
 - 核对用户输入的用户名跟密码
 - 忘记密码功能（选做）：
 - 注册时让用户选择一个问题并给出回答
 - 忘记密码时只要答对当初的问题，则可以更改密码

```
###Login or Sign up?###
Sign up
###Please set your username###
Mike
###Please set your password###
123456
Signed up !
###Login or Sign up?###
Login
###Please enter your username###
Mike
###Please enter your password###
123456
Login Successfully
###Login or Sign up?###
Login
###Please enter your username###
Amy
###Please enter your password###
1342
Invalid username
###Login or Sign up?###
```

Q&A

You Have
Questions
We Have
Answers

次のうち、ArrayListの説明として正しいものを選びなさい。(3つ選択)

- A. nullは扱えない
- B. 動的な配列として動作する
- C. 重複した値は扱えない
- D. スレッドセーフではない
- E. 値を追加する箇所を制御できる

次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. import java.util.ArrayList;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         ArrayList list = new ArrayList<>();
6.         list.add("A");
7.         list.add(10);
8.         list.add('B');
9.         for (Object obj : list) {
10.             System.out.print(obj);
11.         }
12.     }
13. }
```

- A. 5行目でコンパイルエラーが発生する
- B. 6行目でコンパイルエラーが発生する
- C. 7行目でコンパイルエラーが発生する
- D. 8行目でコンパイルエラーが発生する
- E. 9行目でコンパイルエラーが発生する
- F. 「A10B」と表示される
- G. 実行時に例外がスローされる

次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. import java.util.ArrayList;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         ArrayList<String> list = new ArrayList<>();
6.         list.add("A");
7.         list.add(2, "B");
8.         list.add("C");
9.         list.add("D");
10.        for (String str : list) {
11.            System.out.print(str);
12.        }
13.    }
14. }
```

- A. 「ABCD」と表示される
- B. 「ACBD」と表示される
- C. 「ACDB」と表示される
- D. コンパイルエラーが発生する
- E. 実行時に例外がスローされる

次のプログラムをコンパイル、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. import java.util.ArrayList;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         ArrayList<String> list = new ArrayList<>();
6.         list.add("A");
7.         list.set(0, "B");
8.         list.add("C");
9.         list.set(1, "D");
10.        for (String str : list) {
11.            System.out.print(str);
12.        }
13.    }
14. }
```

- A. 「BD」と表示される
- B. 「AD」と表示される
- C. 「BC」と表示される
- D. 「BCD」と表示される
- E. 「ABCD」と表示される
- F. コンパイルエラーが発生する
- G. 実行時に例外がスローされる

次のプログラムをコンパイルし、実行したときの結果として、正しいものを選びなさい。(1つ選択)

```
1. public class Item {
2.     private int id;
3.     private String name;
4.     public Item(int id, String name) {
5.         this.id = id;
6.         this.name = name;
7.     }
8.     public int getId() {
9.         return id;
10.    }
11.    public String getName() {
12.        return name;
13.    }
14. }
```

```
1. import java.util.HashMap;
2. import java.util.Map;
3.
4. public class Sample {
5.     public static void main(String[] args) {
6.         Map<Integer, Item> map = new HashMap<Integer, Item>();
7.         map.put(1, new Item(1, "A"));
8.         map.put(2, new Item(2, "B"));
9.         map.put(3, new Item(3, "C"));
10.        map.put(1, new Item(1, "A"));
11.        map.put(null, new Item(0, "default"));
12.
13.        System.out.println(map.size());
14.    }
15. }
```

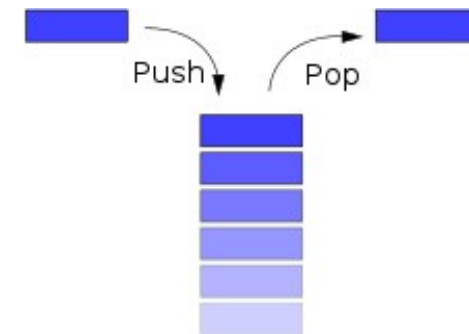
- A. 3が表示される
- B. 4が表示される
- C. 5が表示される
- D. コンパイルエラーが発生する
- E. 実行時に例外がスローされる

Q&A

You Have
Questions
We Have
Answers

练习 时间：15min

- 使用ArrayList实现数据结构栈类[スタック] (Stack)
- 栈的特点：先进后出，后进先出，想象羽毛球桶
- 要求：
 - 该类具有pop方法：删除并返回栈最上面的元素
 - 该类具有push方法：在栈最上面添加新的元素
 - 存int类的栈
- 思考：如何做一个可以存任意类型的栈？



练习 时间：20分钟

- 现有一个char类型的一维数组，里面存了一些字母
- 写一个程序输出每个字母在这个数组里出现的次数
- 数组自己任意提供
- 例：
 - 数组: [f, a, c, b, c, d, a, e, e, d, c, b, d, c]
 - 输出:
a : 2, b : 2, c : 4, d : 3, e : 2, f : 1

Hint: HashMap

练习 时间：10分钟

- 将这段代码8-10行用lambda表达式改写

```
1
2
3 ▶ public class LambdaExpression {
4 ▶     public static void main(String[] args) {
5         int width=10;
6
7         //without lambda, Drawable implementation using anonymous class
8         Drawable d = new Drawable(){
9             public void draw(){System.out.println("Drawing " + width);}
10        };
11        d.draw();
12    }
13 }
14
15 fx interface Drawable{
16     fx public void draw();
17 }
```


练习 时间：30分钟

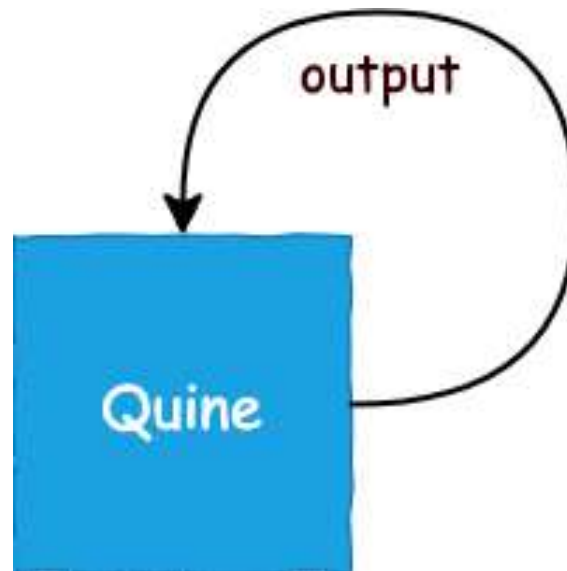
1. 定义一个ThreadTest类，里面有一个非静态变量int count=0。
2. 在ThreadTest类里定义一个add方法，使得每次调用add，count增加1，并输出此时count。
3. 在ThreadTest类里定义一个sub方法，使得每次调用sub，count减少1，并输出此时count。
4. 在ThreadTest类的主方法里创建一个ThreadTest的对象threadTest。
5. 在ThreadTest类的主方法里建立四个线程，其中两个线程每次对threadTest的count加1，另外两个线程每次对threadTest的count减1。

输出结果：

```
Thread-0add:i=1  
Thread-3sub:i=0  
Thread-2sub:i=-1  
Thread-1add:i=0
```

练习 时间：20分钟

- 创建一个类名为Quine.java的文件。
- Quine.java的功能要求读取自身的内容并输出到标准输出
- 请使用java.io API。



THANK YOU