

## Reduce\_JAR\_Size:\_Replacing\_classes\_with\_arrays

*This Article has been submitted by [user:atiskov](#) as a part of the St Petersburg Developer Day competition.*

---

This article is an extension of [Reduce JAR Size](#).

Here you can see two classes with some trivial logic. Lets ?inline? class InlineMe.

```
public class Main {
    private static final int LEN = 6;

    public static void main(String[] args) {
        InlineMe ims[] = new InlineMe[LEN];

        for (int i = 0; i < ims.length; i++) {
            ims[i] = new InlineMe();
            InlineMe im = ims[i];
            im.init(i * 2, i + 1);
            im.setB(i % 2 == 0);
        }

        for (int i = 0; i < ims.length; i++) {
            InlineMe im = ims[i];
            if (im.isB())
                im.print();
        }
    }
}

public class InlineMe {
    private boolean b = false;
    private int ints[] = null;

    public boolean isB() {
        return b;
    }

    public void setB(boolean b) {
        this.b = b;
    }

    protected int[] init(int r, int len){
        ints = new int[len];
        for (int i = 0; i < ints.length; i++) {
            ints[i] = r;
        }
        return ints;
    }

    public void print(){
        for (int i = 0; i < ints.length; i++) {
            int anInt = ints[i];
            System.out.print("anInt = " + anInt+ " ");
        }
        System.out.println("");
    }
}
```

The output result of launching such application is:

```
anInt = 0
anInt = 4 anInt = 4 anInt = 4
```

## Reduce\_JAR\_Size:\_Replacing\_classes\_with\_arrays

anInt = 8 anInt = 8 anInt = 8 anInt = 8 anInt = 8

---

These are steps to do in order to replace class with an arrays:

1. Set `public` modifiers to all members of class `InlineMe`
2. Move all methods of class `InlineMe` into class `Main` with adding parameter of type `InlineMe` into them and setting modifier `static` to these new methods.
3. Replace invoking methods of `InlineMe` by these new methods of class `Main` (with adding appropriate parameter).

You will see something like this:

```
public class Main {
    private static final int LEN = 6;

    public static void main(String[] args) {
        InlineMe ims[] = new InlineMe[LEN];

        for (int i = 0; i < ims.length; i++) {
            ims[i] = new InlineMe();
            InlineMe im = ims[i];
            init(i * 2, i + 1, im);
            setB(i % 2 == 0, im);
        }

        for (int i = 0; i < ims.length; i++) {
            InlineMe im = ims[i];
            if (isB(im))
                print(im);
        }
    }

    public static boolean isB(InlineMe im) {
        return im.b;
    }

    public static void setB(boolean b, InlineMe im) {
        im.b = b;
    }

    protected static int[] init(int r, int len, InlineMe im) {
        im.ints = new int[len];
        for (int i = 0; i < im.ints.length; i++) {
            im.ints[i] = r;
        }
        return im.ints;
    }

    public static void print(InlineMe im) {
        for (int i = 0; i < im.ints.length; i++) {
            int anInt = im.ints[i];
            System.out.print("anInt = " + anInt + " ");
        }
        System.out.println("");
    }
}
```

## Reduce\_JAR\_Size:\_Replacing\_classes\_with\_arrays

```
public class InlineMe {
    boolean b = false;
    int ints[] = null;
}
```

Next steps are:

4. Replace declarations of variables of type InlineMe by instant variables of replaced class InlineMe, adding square brackets to them (making them to be array)
5. Replace parameters of methods, that were moved from InlineMe, which types are InlineMe, by the sequence of those previously added variables in n.4. Add special parameter ?pointer? that will points to the current processed element of an array.
6. Change those methods in the places of their invoking appropriately.

The final result will be:

```
public class Main {
    private static final int LEN = 6;

    public static void main(String[] args) {
        boolean b[] = new boolean[LEN];
        for (int i = 0; i < b.length; i++)
            b[i] = false;
        int ints[][] = new int[LEN][];
        for (int i = 0; i < b.length; i++) {
            init(i * 2, i + 1, b,ints,i);
            setB(i % 2 == 0, b,ints,i);
        }

        for (int i = 0; i < b.length; i++) {
            if (isB(b,ints,i))
                print(b,ints,i);
        }
    }

    public static boolean isB(boolean b[], int ints[][], int pointer) {
        return b[pointer];
    }

    public static void setB(boolean new_b, boolean b[], int ints[][], int pointer) {
        b[pointer] = new_b;
    }

    protected static int[] init(int r, int len, boolean b[], int ints[][], int pointer) {
        ints[pointer] = new int[len];
        for (int i = 0; i < ints[pointer].length; i++) {
            ints[pointer][i] = r;
        }
        return ints[pointer];
    }

    public static void print(boolean b[], int ints[][], int pointer) {
        for (int i = 0; i < ints[pointer].length; i++) {
            int anInt = ints[pointer][i];
            System.out.print("anInt = " + anInt + " ");
        }
    }
}
```

## Reduce\_JAR\_Size:\_Replacing\_classes\_with\_arrays

```
    }  
    System.out.println("");  
  }  
}
```

---