

تمرین هفتم

مدیر مبنا

کلاس انتزاعی زیر را در نظر بگیرید. این کلاس دارای فیلدها و توابع زیر است:

- فیلدهای عدد و مبنا که در سازنده دریافت می‌شوند و توسط تابع `setNumberInBase` نیز قابل تغییرند
- تابع `convertToBase`: یک مبنا دریافت کرده و عدد مربوطه را در آن مبنا برمی‌گرداند
- تابع `sumInBase`: یک عدد و مبنا دریافت کرده و آن را با عدد و مبناي خود جمع کرده و نتیجه را برمی‌گرداند (در هنگام جمع باید مبناها یکسان سازی شوند). نتیجه در مبناي خود شیء برگردانده می‌شود
- تابع `toString`: باید عدد و مبناي جاری را برگرداند.
- تابع `equals`: در صورتی که عدد و مبناي ورودی با عدد و مبناي جاری برابر باشند، مقدار `true` را برمی‌گرداند
- تابع `getInAllBases`: عدد جاری را در مبناي ۲ تا ۱۶ به صورت نشان داده شده در مثال‌ها برمی‌گرداند

```
abstract class BaseManager {
    protected String number;
    protected int base;

    public BaseManager(String number, int base) {
        setNumberInBase(number, base);
    }

    public final void setNumberInBase(String number, int base) {
        this.number = number;
        this.base = base;
    }

    // convert this.number from this.base to newBase and return
    public abstract String convertToBase(int newBase);
    // sum [this.number with this.base] and [number with base] and return the result in this.base
    public abstract String sumInBase(String number, int base);
    // return this.number in base 2..16
    public abstract String getInAllBases();
}
```

کلاس شما باید از کلاس `BaseManager` به ارث برده و توابع انتزاعی را پیاده‌سازی کند. از فایل `Main.java` که همراه با تمرین در اختیار شما قرار گرفته است، برای تست پیاده‌سازی شما استفاده می‌شود. این کلاس دستورات ورودی را گرفته و توابع مربوطه از کلاس `BaseManager` را به صورت نشان داده شده در مثال‌های زیر فراخوانی می‌کند.

نکات:

- مبنا، عددی صحیح بین ۲ تا ۱۶ می باشد.
- اعداد در بازه ۱ تا ۱۰۰۰۰۰ هستند. (بنابراین بیت علامت و اعداد منفی نخواهیم داشت)
- از چاپ کاراکترهای اضافه در خروجی پرهیز کنید!
- کد نوشته شده در کلاس شما باید به شکلی باشد که کلاس Main.java بدون هیچ تغییری در تابع main و BaseManager اجرا شود.
- نام کلاس شما باید دقیقا MyBaseManager باشد.

ورودی اول

```
set 20 10
print
printAll
equal 24 8
equal 25 8
sum 14 16
set AB 16
printAll
```

خروجی اول

```
20 (10)
10100 (02) -- 202 (03) -- 110 (04) -- 40 (05) -- 32 (06) -- 26 (07) -- 24
(08) -- 22 (09) -- 20 (10) -- 19 (11) -- 18 (12) -- 17 (13) -- 16 (14) --
15 (15) -- 14 (16)
TRUE
FALSE
40
10101011 (02) -- 20100 (03) -- 2223 (04) -- 1141 (05) -- 443 (06) -- 333
(07) -- 253 (08) -- 210 (09) -- 171 (10) -- 146 (11) -- 123 (12) -- 102
(13) -- C3 (14) -- B6 (15) -- AB (16)
```

نکات:

- خط دوم در خروجی به دلیل طولانی بودن، در سه خط نمایش داده شده است.
- کلاس Main دستورات زیر را دریافت کرده و توابع متناظر از کلاس BaseManager را فراخوانی می کند:

- | دستور | توضیح |
|-------------|--|
| • set a b | • عدد و مبنای شیء را با فراخوانی تابع setNumberInBase تغییر می‌دهد. چیزی چاپ نمی‌کند! |
| • print | • تابع toString از شیء جاری را فراخوانی کرده و حاصل را چاپ می‌کند. |
| • printAll | • تابع getInAllBases را فراخوانی کرده و حاصل را چاپ می‌کند. |
| • equal a b | • اگر شیء جاری با عدد a در مبنای b برابر باشد، TRUE و در غیراینصورت، FALSE چاپ می‌کند. برای مقایسه دو عدد باید ابتدای مبنای آن‌ها را یکسان‌سازی کرد! |
| • sum a b | • شیء جاری را با عدد a در مبنای b جمع کرده و حاصل را که توسط تابع sumInBase بازگردانده می‌شود، چاپ می‌کند. |