

Esame di Programmazione II, 18 giugno 2015

La libreria Java standard contiene un'interfaccia che specifica una sequenza di caratteri di cui è possibile estrarre sottosequenze:

```
public interface java.lang.CharSequence {
    public abstract int length();
    public abstract char charAt(int pos);
    public abstract java.lang.CharSequence subSequence(int start, int end);
    public abstract java.lang.String toString();
}
```

Il metodo `length()` restituisce il numero di caratteri della sequenza; il metodo `charAt()` restituisce il `pos`-esimo carattere della sequenza e genera una `IndexOutOfBoundsException` se `pos` è negativo o maggiore o uguale di `length()`; il metodo `subSequence()` ritorna una sottosequenza dal carattere `start`-esimo (incluso) al carattere `end`-esimo (escluso) e genera una `IndexOutOfBoundsException` se `start` o `end` sono negativi o se `end` è maggiore di `length()` o se `start` è maggiore di `end`; il metodo `toString()` ritorna una stringa formata dalla concatenazione dei caratteri della sequenza. Si noti che `java.lang.String` implementa `java.lang.CharSequence`.

Esercizio 1 [4 punti] Si implementi una classe astratta `AbstractCharSequence` che implementa `CharSequence` lasciandone tutti i metodi astratti, tranne `toString()` che viene implementato e specificato come `final`.

Esercizio 2 [9 punti] Si completi la seguente sottoclasse di `AbstractCharSequence`, che specifica una sequenza di caratteri alfabetici che parte da un dato carattere `start` e continua con i successivi caratteri minuscoli dell'alfabeto inglese, ciclicamente, per una certa lunghezza specificata al costruttore: dopo la 'z' si ricomincia dalla 'a'. Nel costruttore, `start` deve essere tra 'a' e 'z' altrimenti deve venire lanciata una `IllegalArgumentException`; il parametro `length` deve essere non negativo, altrimenti deve venire lanciata una `IndexOutOfBoundsException`:

```
public class Alphabetical extends AbstractCharSequence {
    ...
    public Alphabetical(char start, int length) { ... }

    @Override
    public int length() { ... }

    @Override
    public char charAt(int index) { ... }

    @Override
    public Alphabetical subSequence(int start, int end) { ... }
}
```

Si noti che il tipo di ritorno del metodo `subSequence()` è stato raffinato e questo dovrà venire rispettato dalla sua implementazione.

Esercizio 3 [9 punti] Si completi la seguente sottoclasse di `AbstractCharSequence`, che permette di costruire una sequenza identica a quella fonita al suo costruttore, ma con in più un carattere di controllo alfabetico in fondo. Tale carattere è definito come la somma dei codici dei caratteri della sequenza originale, modulo 26, più il codice di 'a'.

```
public class ControlCode extends AbstractCharSequence {
    ...
    public ControlCode(CharSequence original) { ... }

    @Override
    public int length() { ... }

    @Override
    public char charAt(int index) { ... }

    @Override
    public CharSequence subSequence(int start, int end) { ... }
}
```

Se tutto è corretto, l'esecuzione del seguente programma:

```
public class Main {

    private static int counter;

    public static void main(String[] args) {
        CharSequence a1 = new Alphabetical('c', 26);
        CharSequence a2 = new Alphabetical('h', 60);
        CharSequence a3 = a2.subSequence(5, 40);
        print(a1); // 1
        print(a2); // 2
        print(a3); // 3

        CharSequence c1 = new ControlCode("Shakespeare");
        CharSequence c2 = new ControlCode(a3);
        print(c1); // 4
        print(c2); // 5

        CharSequence s1 = "There is no world without Verona walls";
        CharSequence s2 = "But purgatory, torture, hell itself";
        print(s1); // 6
        print(s2); // 7

        CharSequence app = s1.toString() + a2.toString();
        print(app); // 8

        CharSequence last = c2.subSequence(35, 36);
        print(last); // 9
    }

    private static void print(CharSequence seq) {
        System.out.println(++counter + ": " + seq + " length = " + seq.length());
    }
}
```

dovrà stampare:

```
1: cdefghijklmnopqrstuvwxyzab length = 26
2: hijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnop length = 60
3: mnopqrstuvwxyzabcdefghijklmnopqrstu length = 35
4: Shakespeareo length = 12
5: mnopqrstuvwxyzabcdefghijklmnopqrstuq length = 36
6: There is no world without Verona walls length = 38
7: But purgatory, torture, hell itself length = 35
8: There is no world without Verona wallshijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnop length = 98
9: q length = 1
```