

# **Programovanie (4)**

## **Cvičenie 01 - Code review**

**Alexander Šimko**

- Code review je subjektívna záležitosť
- Ak by ju robil niekto iný, mal by na kód iný názor

# Aký kód písať?

1. Píšte kód, ktorý funguje správne (bez chýb, bez memory leakov)
2. Píšte čitateľný kód
3. Píšte efektívny kód (časová, pamäťová zložitosť)

**ISBN po 1.**

# Čo je nedobré s touto metódou?

```
public static boolean isbn(String isbn){
    isbn = isbn.replace("ISBN-10", "");
    isbn = isbn.replace("ISBN-13", "");
    isbn = isbn.replace("ISBN", "");
    isbn = isbn.replace(":", "");
    isbn = isbn.replace("-", "");
    isbn = isbn.replace(" ", "");
    if (isbn.length() == 10 && ((isbn.contains("x") && (isbn.indexOf("x")!=9)) ||
        (isbn.contains("x") && isbn.indexOf("x")!=9))){
        return false;
    }
    if (isbn.length() != 10 && (isbn.contains("x") || isbn.contains("x"))){
        return false;
    }

    int sum = 0;
    if (isbn.length()==10){
        int i = 1;
        if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x'){
            sum += 10;
            isbn = isbn.substring(0,9);
            i ++;
        }
        long isbnnum = Long.parseLong(isbn);

        while (isbnnum > 0) {
            sum += isbnnum % 10 * i;
            isbnnum /= 10;
            ++i;
        }
        return sum % 11 == 0;
    }
    if (isbn.length() == 13){
        int i = 1;
        long isbnnum = Long.parseLong(isbn.substring(0,12));
        while (isbnnum > 0) {
            if (i % 2 == 0) {
                sum += isbnnum % 10;
            }
            else{
                sum += isbnnum % 10 * 3;
            }
            isbnnum /= 10;
            ++i;
        }
        if (sum % 10 == 0){
            return true;
        }
        return (10- sum % 10) == (isbn.toCharArray()[12]-'0');
    }
    return false;
}
```

# Čo je nedobré s touto metódou?

```
public static boolean isbn(String isbn){
    isbn = isbn.replace("ISBN-10", "");
    isbn = isbn.replace("ISBN-13", "");
    isbn = isbn.replace("ISBN", "");
    isbn = isbn.replace(":", "");
    isbn = isbn.replace("-", "");
    isbn = isbn.replace(" ", "");
    if (isbn.length() == 10 && ((isbn.contains("x") && (isbn.indexOf("x")!=9)) ||
        (isbn.contains("x") && isbn.indexOf("x")!=9))){
        return false;
    }
    if (isbn.length() != 10 && (isbn.contains("x") || isbn.contains("x"))){
        return false;
    }

    int sum = 0;
    if (isbn.length()==10){
        int i = 1;
        if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x'){
            sum += 10;
            isbn = isbn.substring(0,9);
            i ++;
        }
        long isbnnum = Long.parseLong(isbn);

        while (isbnnum > 0) {
            sum += isbnnum % 10 * i;
            isbnnum /= 10;
            ++i;
        }
        return sum % 11 == 0;
    }
    if (isbn.length() == 13){
        int i = 1;
        long isbnnum = Long.parseLong(isbn.substring(0,12));
        while (isbnnum > 0) {
            if (i % 2 == 0) {
                sum += isbnnum % 10;
            }
            else{
                sum += isbnnum % 10 * 3;
            }
            isbnnum /= 10;
            ++i;
        }
        if (sum % 10 == 0){
            return true;
        }
        return (10- sum % 10) == (isbn.toCharArray()[12]-'0');
    }
    return false;
}
```

Kombinuje viac separátnych vecí,  
a preto je dlhá

# Čo je nedobré s touto metódou?

```
public static boolean isbn(String isbn){
    isbn = isbn.replace("ISBN-10", "");
    isbn = isbn.replace("ISBN-13", "");
    isbn = isbn.replace("ISBN", "");
    isbn = isbn.replace(":", "");
    isbn = isbn.replace("-", "");
    isbn = isbn.replace(" ", "");
    if (isbn.length() == 10 && ((isbn.contains("x") && isbn.contains("X")) || isbn.contains("0")))
        return false;
    }
    if (isbn.length() != 10 && (isbn.contains("x") || isbn.contains("X")))
        return false;
    }

    int sum = 0;
    if (isbn.length() == 10){
        int i = 1;
        if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'X'){
            sum += 10;
            isbn = isbn.substring(0, 9);
            i++;
        }
        long isbnnum = Long.parseLong(isbn);

        while (isbnnum > 0) {
            sum += isbnnum % 10 * i;
            isbnnum /= 10;
            ++i;
        }
        return sum % 11 == 0;
    }
    if (isbn.length() == 13){
        int i = 1;
        long isbnnum = Long.parseLong(isbn);
        while (isbnnum > 0) {
            if (i % 2 == 0) {
                sum += isbnnum % 10;
            }
            else{
                sum += isbnnum % 10 * 3;
            }
            isbnnum /= 10;
            ++i;
        }
        if (sum % 10 == 0){
            return true;
        }
        return (10 - sum % 10) == (isbn.charAt(12) - '0');
    }
    return false;
}
```

```
public static boolean isbn(String isbn){
    String numbers = extractNumbersFrom(isbn);

    if (numbers.length() == 10) {
        return isValidIsbn10(numbers);
    }
    if (numbers.length() == 13) {
        return isValidIsbn13(numbers);
    }
    return false;
}

private static String extractNumbersFrom(String isbn) {
    isbn = isbn.replace("ISBN-10", "");
    isbn = isbn.replace("ISBN-13", "");
    isbn = isbn.replace("ISBN", "");
    isbn = isbn.replace(":", "");
    isbn = isbn.replace("-", "");
    isbn = isbn.replace(" ", "");
    return isbn;
}

private static boolean isValidIsbn10(String isbn) {
    if (((isbn.contains("x") && (isbn.indexOf("x") != 9)) || (isbn.contains("X") && isbn.indexOf("X") != 9))) {
        return false;
    }

    int sum = 0;
    int i = 1;
    if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'X'){
        sum += 10;
        isbn = isbn.substring(0, 9);
        i++;
    }
    long isbnnum = Long.parseLong(isbn);

    while (isbnnum > 0) {
        sum += isbnnum % 10 * i;
        isbnnum /= 10;
        ++i;
    }
    return sum % 11 == 0;
}
```

# Čo je nevhodné tu?

```
private static boolean isValidIsbn10(String isbn) {
    if (((isbn.contains("x") && (isbn.indexOf("x")!=9)) ||
        (isbn.contains("x") && isbn.indexOf("x")!=9))) {
        return false;
    }

    int sum = 0;
    int i = 1;
    if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x') {
        sum += 10;
        isbn = isbn.substring(0,9);
        i ++;
    }
    long isbnnum = Long.parseLong(isbn);

    while (isbnnum > 0) {
        sum += isbnnum % 10 * i;
        isbnnum /= 10;
        ++i;
    }
    return sum % 11 == 0;
}
```



# Čo je nevhodné tu?

```
private static boolean isValidIsbn10(String isbn) {  
    if (( (isbn.contains("x") && (isbn.indexOf("x") != 9)) |  
          (isbn.contains("x") && isbn.indexOf("x") != 9)) ) {  
        return false;  
    }  
  
    int sum = 0;  
    int i = 1;  
    if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x') {  
        sum += 10;  
        isbn = isbn.substring(0, 9);  
        i++;  
    }  
    long isbnnum = Long.parseLong(isbn);  
  
    while (isbnnum > 0) {  
        sum += isbnnum % 10 * i;  
        isbnnum /= 10;  
        ++i;  
    }  
    return sum % 11 == 0;  
}
```

Ten istý znak hľadáme 2x

# Čo je nevhodné tu?

```
private static boolean isValidIsbn10(String isbn) {  
    if (( (isbn.contains("x") && (isbn.indexOf("x") != 9)) |  
          (isbn.contains("x") && isbn.indexOf("x") != 9)) ) {  
        return false;  
    }  
}
```

Ten istý znak hľadáme 2x

```
int sum = 0;  
int i = 1;  
if (isbn.toCharArray()[9] {  
    sum += 10;  
    isbn = isbn.substring(0, 9);  
    i++;  
}  
long isbnnum = Long.parseLong(isbn);  
  
while (isbnnum > 0) {  
    sum += isbnnum % 10 * i;  
    isbnnum /= 10;  
    ++i;  
}  
return sum % 11 == 0;  
}
```

```
int indexOfSmallX = isbn.indexOf("x");  
if (indexOfSmallX != -1 && indexOfSmallX != 9) {  
    return false;  
}  
int indexOfBigX = isbn.indexOf("X");  
if (indexOfBigX != -1 && indexOfBigX != 9) {  
    return false;  
}
```

# Čo je nevhodné tu?

```
private static boolean isValidIsbn10(String isbn) {  
    if (( (isbn.contains("x") && (isbn.indexOf("x") != 9)) |  
          (isbn.contains("x") && isbn.indexOf("x") != 9)) ) {  
        return false;  
    }  
}
```

Ten istý znak hľadáme 2x

```
int sum = 0;  
int i = 1;  
if (isbn.toCharArray()[9] {  
    sum += 10;  
    isbn = isbn.substring(0, 9);  
    i++;  
}  
long isbnnum = Long.parseLong(isbn);  
  
while (isbnnum > 0) {  
    sum += isbnnum % 10 * i;  
    isbnnum /= 10;  
    ++i;  
}  
return sum % 11 == 0;  
}
```

```
int indexOfSmallX = isbn.indexOf("x");  
if (indexOfSmallX != -1 && indexOfSmallX != 9)  
    return false;  
  
int indexOfBigX = isbn.indexOf("X");  
if (indexOfBigX != -1 && indexOfBigX != 9) {  
    return false;  
}
```

Stačí hľadať znak, nie reťazec

# Čo je nevhodné tu?

```
private static boolean isValidIsbn10(String isbn) {  
    if (( (isbn.contains("x") && (isbn.indexOf("x") != 9)) |  
          (isbn.contains("x") && isbn.indexOf("x") != 9)) ) {  
        return false;  
    }  
}
```

Ten istý znak hľadáme 2x

```
int sum = 0;  
int i = 1;  
if (isbn.toCharArray()[9] {  
    sum += 10;  
    isbn = isbn.substring(0, 9);  
    i++;  
}  
long isbnnum = Long.parseLong(isbn);  
  
while (isbnnum > 0) {  
    sum += isbnnum % 10 * i;  
    isbnnum /= 10;  
    ++i;  
}  
return sum % 11 == 0;  
}
```

```
int indexOfSmallX = isbn.indexOf("x");  
if (indexOfSmallX != -1 && indexOfSmallX != 9) {  
    return false;  
}  
int indexOfBigX = isbn.indexOf("x");  
if (indexOfBigX != -1 && indexOfBigX != 9) {  
    return false;  
}
```

Stačí hľadať znak, nie reťazec

```
int indexOfSmallX = isbn.indexOf('x');  
if (indexOfSmallX != -1 && indexOfSmallX != 9) {  
    return false;  
}  
int indexOfBigX = isbn.indexOf('x');  
if (indexOfBigX != -1 && indexOfBigX != 9) {  
    return false;  
}
```

# Čo ďalšie?

```
private static boolean isValidIsbn10(String isbn) {
    if (((isbn.contains("x") && (isbn.indexOf("x") != 9)) ||
        (isbn.contains("x") && isbn.indexOf("x") != 9))) {
        return false;
    }

    int sum = 0;
    int i = 1;
    if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x') {
        sum += 10;
        isbn = isbn.substring(0, 9);
        i++;
    }
    long isbnnum = Long.parseLong(isbn);

    while (isbnnum > 0) {
        sum += isbnnum % 10 * i;
        isbnnum /= 10;
        ++i;
    }
    return sum % 11 == 0;
}
```

# Čo ďalšie?

```
private static boolean isValidIsbn10(String isbn) {  
    if (((isbn.contains("x") && (isbn.indexOf("x") != 9)) ||  
        (isbn.contains("x") && isbn.indexOf("x") != 9))) {  
        return false;  
    }  
  
    int sum = 0;  
    int i = 1;  
    if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x') {  
        sum += 10;  
        isbn = isbn.substring(0, 9);  
        i++;  
    }  
    long isbnnum = Long.parseLong(isbn);  
  
    while (isbnnum > 0) {  
        sum += isbnnum % 10 * i;  
        isbnnum /= 10;  
        ++i;  
    }  
    return sum % 11 == 0;  
}
```

Dynamická alokácia dvoch zbytočných polí

# Javadoc je váš kamarát

```
private static boolean isValidIsbn10(String isbn) {  
    if (((isbn.contains("x") && (isbn.indexOf("x") != 9)) ||  
        (isbn.contains("x") && isbn.indexOf("x") != 9))) {  
        return false;  
    }  
  
    int sum = 0;  
    int i = 1;  
    if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x') {  
        sum += 10;  
        isbn = isbn.substring(0, 9);  
        i++;  
    }  
    long isbnnum = 0;  
  
    while (isbnnum < 10) {  
        sum += isbnnum;  
        isbnnum /= 10;  
        ++i;  
    }  
    return sum % 11 == 0;  
}
```

Dynamická alokácia dvoch zbytočných polí

The screenshot shows the Oracle Java SE 17 & JDK 17 API documentation for the `String.toCharArray()` method. The browser address bar shows `docs.oracle.com/en/java/javase/17/docs/api/java.base/java`. The navigation bar includes tabs for OVERVIEW, MODULE, PACKAGE, CLASS (selected), USE, TREE, PREVIEW, NEW, DEPRECATED, INDEX, and HELP. Below the navigation bar, there are links for SUMMARY: NESTED, FIELD, CONSTR, METHOD and DETAIL: FIELD, CONSTR, METHOD. A search bar is visible with the text "SEARCH: Search". The main content area shows the method signature `public char[] toCharArray()` and its description: "Converts this string to a new character array." Under the "Returns:" section, it states: "a newly allocated character array whose length is the length of this string and whose contents are initialized to contain the character sequence represented by this string."

# Čo ďalšie?

```
private static boolean isValidIsbn10(String isbn) {  
    if (((isbn.contains("x") && (isbn.indexOf("x") != 9)) ||  
        (isbn.contains("x") && isbn.indexOf("x") != 9))) {  
        return false;  
    }  
  
    int sum = 0;  
    int i = 1;  
    if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x') {  
        sum += 10;  
        isbn = isbn.substring(0, 9);  
        i++;  
    }  
    long isbnnum = Long.parseLong(isbn);  
    while (isbnnum > 0) {  
        sum += isbnnum % 10 * i;  
        isbnnum /= 10;  
        ++i;  
    }  
    return sum % 11 == 0;  
}
```

Dynamická alokácia dvoch zbytočných polí

```
char lastChar = isbn.charAt(9);  
if (lastChar == 'x' || lastChar == 'x') {  
    ...  
}
```



# A ešte jedna vec

```
private static boolean isValidIsbn10(String isbn) {
    if (((isbn.contains("x") && (isbn.indexOf("x")!=9)) ||
        (isbn.contains("x") && isbn.indexOf("x")!=9))) {
        return false;
    }

    int sum = 0;
    int i = 1;
    if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x') {
        sum += 10;
        isbn = isbn.substring(0,9);
        i ++;
    }
    long isbnnum = Long.parseLong(isbn);

    while (isbnnum > 0) {
        sum += isbnnum % 10 * i;
        isbnnum /= 10;
        ++i;
    }
    return sum % 11 == 0;
}
```

# A ešte jedna vec

```
private static boolean isValidIsbn10(String isbn) {  
    if (((isbn.contains("x") && (isbn.indexOf("x") != 9)) ||  
        (isbn.contains("x") && isbn.indexOf("x") != 9))) {  
        return false;  
    }  
}
```

Špeciálne riešime posledné X,  
ťažšie na pochopenie

```
int sum = 0;
```

```
int i = 1;
```

```
if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x') {  
    sum += 10;  
    isbn = isbn.substring(0, 9);  
    i++;  
}
```

```
long isbnnum = Long.parseLong(isbn);
```

```
while (isbnnum > 0) {  
    sum += isbnnum % 10 * i;  
    isbnnum /= 10;  
    ++i;  
}
```

```
return sum % 11 == 0;
```

```
}
```

# A ešte jedna vec

```
private static boolean isValidIsbn10(String isbn) {  
    if (((isbn.contains("x") && (isbn.indexOf("x") != 9)) ||  
        (isbn.contains("X") && isbn.indexOf("X") != 9))) {  
        return false;  
    }  
}
```

Špeciálne riešime posledné X,  
ťažšie na pochopenie

```
int sum = 0;  
int i = 1;
```

```
if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'X') {  
    sum += 10;  
    isbn = isbn.substring(0, 9);  
    i++;  
}
```

```
long isbnnum = Long.parseLong(isbn);
```

```
while (isbnnum > 0) {  
    sum += isbnnum % 10 * i;  
    isbnnum /= 10;  
    ++i;  
}
```

```
return sum % 11 == 0;
```

```
}
```

```
private int charToInt(char ch) {  
    if (ch == 'x' || ch == 'X') {  
        return 10;  
    }  
    return ch - '0';  
}  
  
int length = isbn.length();  
int sum = 0;  
for (int i = 0; i < length; ++i) {  
    int digit = charToInt(isbn.charAt(i));  
    int weight = 10 - i;  
    sum += digit * weight;  
}  
  
return sum % 11 == 0;
```

# Stará verzia

```
private static boolean isValidIsbn10(String isbn) {
    if (((isbn.contains("x") && (isbn.indexOf("x")!=9)) ||
        (isbn.contains("x") && isbn.indexOf("x")!=9))) {
        return false;
    }

    int sum = 0;
    int i = 1;
    if (isbn.toCharArray()[9] == 'x' || isbn.toCharArray()[9] == 'x') {
        sum += 10;
        isbn = isbn.substring(0,9);
        i ++;
    }
    long isbnnum = Long.parseLong(isbn);

    while (isbnnum > 0) {
        sum += isbnnum % 10 * i;
        isbnnum /= 10;
        ++i;
    }
    return sum % 11 == 0;
}
```

# Stará a nová verzia

```
private static boolean isValidIsbn10(String isbn) {
```

```
    if (((isbn.contains('x') || isbn.contains('X')) && isbn.length() != 10) ||
        (isbn.charAt(0) != '0' && isbn.charAt(0) != '3'))
        return false;
}
```

```
int sum = 0;
int i = 1;
if (isbn.toCharArray().length > 10)
    sum += 10;
isbn = isbn.substring(1);
i++;
```

```
}
long isbnnum = Long.parseLong(isbn);
```

```
while (isbnnum > 0)
    sum += isbnnum % 10;
    isbnnum /= 10;
    ++i;
```

```
}
return sum % 11 == 0;
```

```
}
```

```
private int charToInt(char ch) {
    if (ch == 'x' || ch == 'X') {
        return 10;
    }
    return ch - '0';
}
```

```
}
```

```
private static boolean isValidIsbn10(String isbn) {
    int indexOfSmallX = isbn.indexOf("x");
    if (indexOfSmallX != -1 && indexOfSmallX != 9) {
        return false;
    }
```

```
int indexOfBigX = isbn.indexOf("X");
if (indexOfBigX != -1 && indexOfBigX != 9) {
    return false;
}
```

```
int length = isbn.length();
int sum = 0;
for (int i = 0; i < length; ++i) {
    int digit = charToInt(isbn.charAt(i));
    int weight = 10 - i;
    sum += digit * weight;
}
```

```
return sum % 11 == 0;
```

```
}
```

# Čo je zlé tu?

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }
    int i = 1;
    long isbnnum = Long.parseLong(isbn.substring(0,12));
    while (isbnnum > 0) {
        if (i % 2 == 0) {
            sum += isbnnum % 10;
        }
        else{
            sum += isbnnum % 10 * 3;
        }
        isbnnum /= 10;
        ++i;
    }
    if (sum % 10 == 0){
        return true;
    }
    return (10- sum % 10) == (isbn.toCharArray()[12]-'0');
}
```

# Čo je zlé tu?

```
private static boolean isValidIsbn13(String isbn) {  
    if (isbn.contains("x") || isbn.contains("X")){  
        return false;  
    }  
    int i = 1;  
    long isbnnum = Long.parseLong(isbn.substring(0,12));  
    while (isbnnum > 0) {  
        if (i % 2 == 0) {  
            sum += isbnnum % 10;  
        }  
        else{  
            sum += isbnnum % 10 * 3;  
        }  
        isbnnum /= 10;  
        ++i;  
    }  
    if (sum % 10 == 0){  
        return true;  
    }  
    return (10 - sum % 10) == (isbn.toCharArray()[12] - '0');  
}
```

Duplicita

# Čo je zlé tu?

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }
    int i = 1;
    long isbnnum = Long.parseLong(isbn.substring(0,12));
    while (isbnnum > 0) {
        if (i % 2 == 0) {
            sum += isbnnum % 10;
        }
        else{
            sum += isbnnum % 10 * 3;
        }
        isbnnum /= 10;
        ++i;
    }
    if (sum % 10 == 0){
        return true;
    }
    return (10 - sum % 10) == (isbn.toCharArray()[12] - '0');
}
```

Duplicita

```
int digit = isbnnum % 10;
int weight = i % 2 == 0 ? 1 : 3;
sum += digit * weight;
```



# Čo ďalšie?

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }
    int i = 1;
    long isbnnum = Long.parseLong(isbn.substring(0,12));
    while (isbnnum > 0) {
        if (i % 2 == 0) {
            sum += isbnnum % 10;
        }
        else{
            sum += isbnnum % 10 * 3;
        }
        isbnnum /= 10;
        ++i;
    }
    if (sum % 10 == 0){
        return true;
    }
    return (10- sum % 10) == (isbn.toCharArray()[12]-'0');
}
```

# Čo ďalšie?

```
private static boolean isValidIsbn13(String isbn) {  
    if (isbn.contains("x") || isbn.contains("X")){  
        return false;  
    }  
    int i = 1;  
    long isbnnum = Long.parseLong(isbn.substring(0, 12));  
    while (isbnnum > 0) {  
        if (i % 2 == 0) {  
            sum += isbnnum % 10;  
        }  
        else{  
            sum += isbnnum % 10 * 3;  
        }  
        isbnnum /= 10;  
        ++i;  
    }  
    if (sum % 10 == 0){  
        return true;  
    }  
    return (10 - sum % 10) == (isbn.toCharArray()[12] - '0');  
}
```

**Zbytočná alokácia nového reťazca**

# Čo ďalšie?

```
private static boolean isValidIsbn13(String isbn) {  
    if (isbn.contains("x") || isbn.contains("X")){  
        return false;  
    }  
    int i = 1;  
    long isbnnum = Long.parseLong(isbn.substring(0,12));  
    while (isbnnum > 0) {  
        if (i % 2 == 0) {  
            sum += isbnnum % 10;  
        }  
        else{  
            sum += isbnnum % 10 * 3;  
        }  
        isbnnum /= 10;  
        ++i;  
    }  
    if (sum % 10 == 0){  
        return true;  
    }  
    return (10 - sum % 10) == (isbn.toCharArray()[12] - '0');  
}
```

**Zbytočná alokácia nového reťazca**

```
for (int i = 0; i < 12; ++i) {  
    int digit = charToInt(isbn.charAt(i));  
    ...  
}
```

# Už sme to mali...

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }
    int i = 1;
    long isbnnum = Long.parseLong(isbn.substring(0,12));
    while (isbnnum > 0) {
        if (i % 2 == 0) {
            sum += isbnnum % 10;
        }
        else{
            sum += isbnnum % 10 * 3;
        }
        isbnnum /= 10;
        ++i;
    }
    if (sum % 10 == 0){
        return true;
    }
    return (10- sum % 10) == (isbn.toCharArray()[12]-'0');
}
```

# Už sme to mali...

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }
    int i = 1;
    long isbnnum = Long.parseLong(isbn.substring(0,12));
    while (isbnnum > 0) {
        if (i % 2 == 0) {
            sum += isbnnum % 10;
        }
        else{
            sum += isbnnum % 10 * 3;
        }
        isbnnum /= 10;
        ++i;
    }
    if (sum % 10 == 0){
        return true;
    }
    return (10 - sum % 10) == (isbn.toCharArray()[12] - '0');
}
```

**Zbytočná alokácia nového poľa**

# Už sme to mali...

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }
    int i = 1;
    long isbnnum = Long.parseLong(isbn.substring(0,12));
    while (isbnnum > 0) {
        if (i % 2 == 0) {
            sum += isbnnum % 10;
        }
        else{
            sum += isbnnum % 10 * 3;
        }
        isbnnum /= 10;
        ++i;
    }
    if (sum % 10 == 0){
        return true;
    }
    return (10 - sum % 10) == (isbn.toCharArray()[12] - '0');
```

**Zbytočná alokácia nového poľa**

```
return (10 - sum % 10) == charToInt(isbn.charAt(12));
```

# A ešte jedna vec

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }
    int i = 1;
    long isbnnum = Long.parseLong(isbn.substring(0,12));
    while (isbnnum > 0) {
        if (i % 2 == 0) {
            sum += isbnnum % 10;
        }
        else{
            sum += isbnnum % 10 * 3;
        }
        isbnnum /= 10;
        ++i;
    }
    if (sum % 10 == 0){
        return true;
    }
    return (10- sum % 10) == (isbn.toCharArray()[12]-'0');
}
```

# A ešte jedna vec

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }
    int i = 1;
    long isbnnum = Long.parseLong(isbn.substring(0,12));
    while (isbnnum > 0) {
        if (i % 2 == 0) {
            sum += isbnnum % 10;
        }
        else{
            sum += isbnnum % 10 * 3;
        }
        isbnnum /= 10;
        ++i;
    }
    if (sum % 10 == 0){
        return true;
    }
    return (10 - sum % 10) == (isbn.toCharArray()[12] - '0');
}
```

**Zlá čitateľnosť**



# Čo je zlé tu?

```
private static boolean isValidIsbn13(String isbn) {  
    if (isbn.contains("x") || isbn.contains("X")){  
        return false;  
    }  
    int i = 1;  
    long isbnnum = Long.parseLong(isbn.substring(0,12));  
    while (isbnnum > 0) {  
        if (i % 2 == 0) {  
            sum += isbnnum % 10;  
        }  
        else{  
            sum += isbnnum % 10 * 3;  
        }  
        isbnnum /= 10;  
        ++i;  
    }  
    if (sum % 10 == 0){  
        return true;  
    }  
    return (10 - sum % 10) == (isbn.toCharArray()[12] - '0');  
}
```

**Zlá čitateľnosť**

```
int remainder = sum % 10;  
int controlDigit = remainder == 0 ? 0 : 10 - remainder;  
  
return controlDigit == charToInt(isbn.charAt(12));
```

# Stará verzia

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }
    int i = 1;
    long isbnnum = Long.parseLong(isbn.substring(0,12));
    while (isbnnum > 0) {
        if (i % 2 == 0) {
            sum += isbnnum % 10;
        }
        else{
            sum += isbnnum % 10 * 3;
        }
        isbnnum /= 10;
        ++i;
    }
    if (sum % 10 == 0){
        return true;
    }
    return (10- sum % 10) == (isbn.toCharArray()[12]-'0');
}
```

# Stará a nová verzia

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }
    int i = 1;
    long isbnnum = Long.parseLong(isbn);
    while (isbnnum > 0)
        if (i % 2 == 0)
            sum += isbnnum;
        else{
            sum += isbnnum;
        }
        isbnnum /= 10;
        ++i;
    }
    if (sum % 10 == 0){
        return true;
    }
    return (10 - sum % 10) % 10 == 0;
}
```

```
private static boolean isValidIsbn13(String isbn) {
    if (isbn.contains("x") || isbn.contains("X")){
        return false;
    }

    for (int i = 0; i < 13; ++i) {
        int digit = charToInt(isbn.charAt(i));
        int weight = i % 2 == 0 ? 1 : 3;
        sum += digit * weight;
    }

    int remainder = sum % 10;
    int controlDigit = remainder == 0 ? 0 : 10 - remainder;

    return controlDigit == charToInt(isbn.charAt(12));
}
```

**ISBN po 2.**

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')){
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')){
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')) {
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')) {
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```

Naozaj netuším na čo je táto premenná...

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')) {
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')) {
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'x')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'x'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'x'){
            return false;
        }
    }
}
```

Naozaj netuším na čo je táto premenná...

Názov premennej má komunikovať účel

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')){
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')){
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```



# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean ciska_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')){
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')){
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                ciska_zacali = true;
            }
        }
        if(ciska_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```

**Dodržiujte mennú konvenciu Javy**

**boolean** digitsStarted = **false**;

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == '1'){
            if ((isbn.charAt(i+1) != 's') ||
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')){
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'x')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'x'){
                fixed += 'x';
            }else {
                fixed += isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'x'){
            return false;
        }
    }
}
```

Dížku stačí zistiť raz, nemusíme v každej iterácii

```
int length = isbn.length();
for (int i = 0 ; i < length; ++i) {
    ...
}
```

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')){
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')){
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean ciska_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')) {
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')) {
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3') {
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                ciska_zacali = true;
            }
        }
        if(ciska_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```

pozícia i+1 nemusí existovať

IndexOutOfBoundsException

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i<isbn.length(); i++){
        if(isbn.charAt(i)=='i'){
            if ((isbn.charAt(i+1)!='s') || (isbn.charAt(i+2)!='b') || (isbn.charAt(i+3)!='n')){
                return false;
            }
        }
        if(isbn.charAt(i)=='I'){
            if ((isbn.charAt(i+1)!='S') || (isbn.charAt(i+2)!='B') || (isbn.charAt(i+3)!='N')){
                return false;
            }
            if(isbn.charAt(i+4)=='-' && isbn.charAt(i+5)=='1' && isbn.charAt(i+6)=='3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i)>=48 && isbn.charAt(i)<=57) || (isbn.charAt(i)=='x') ||
(isbn.charAt(i)=='X')){
            if(isbn.charAt(i)=='x' || isbn.charAt(i)=='X'){
                fixed+='x';
            }else {
                fixed+=isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i)<48 || isbn.charAt(i)>57) && isbn.charAt(i)!='-'
&& isbn.charAt(i)!=' ' && isbn.charAt(i)!='x' && isbn.charAt(i)!='X'){
            return false;
        }
    }
}
```

Magické konštanty

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')) {
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')) {
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3') {
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'x')){
            if(isbn.charAt(i) == '0' || isbn.charAt(i) == '9')
                fixed += 'x';
            }else {
                fixed += isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'x'){
            return false;
        }
    }
}
```

Magické konštanty

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i<isbn.length(); i++){
        if(isbn.charAt(i)=='i'){
            if ((isbn.charAt(i+1)!='s') || (isbn.charAt(i+2)!='b') || (isbn.charAt(i+3)!='n')){
                return false;
            }
        }
        if(isbn.charAt(i)=='I'){
            if ((isbn.charAt(i+1)!='S') || (isbn.charAt(i+2)!='B') || (isbn.charAt(i+3)!='N')){
                return false;
            }
            if(isbn.charAt(i+4)=='-'&&isbn.charAt(i+5)=='1'&&isbn.charAt(i+6)=='3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i)>=48 && isbn.charAt(i)<=57) || (isbn.charAt(i)=='x') ||
(isbn.charAt(i)=='x')){
            if(isbn.charAt(i)=='x' || isbn.charAt(i)=='x'){
                fixed+='x';
            }else {
                fixed+=isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i)<48 || isbn.charAt(i)>57) && isbn.charAt(i)!='-'
&& isbn.charAt(i)!=' ' && isbn.charAt(i)!='x'&&isbn.charAt(i)!='x'){
            return false;
        }
    }
}
```

# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')){
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')){
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```

To isté volanie 6x



# Čo vidíme tu?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')) {
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')) {
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3') {
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                char ch = isbn.charAt(i);
                if ((ch >= '0' && ch <= '9') || ch == 'x' || ch == 'X') {
                    ...
                }
            }
        }
        if(cisla_zacali && isbn.charAt(i) != ' '){
            return false;
        }
    }
}
```

To isté volanie 6x

# O čo sa ale snaží tento kód?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean cisla_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')){
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')){
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                cisla_zacali = true;
            }
        }
        if(cisla_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```

# O čo sa ale snaží tento kód?

Kód je ťažko zrozumiteľný

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean ciska_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')) {
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')) {
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                ciska_zacali = true;
            }
        }
        if(ciska_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```

# O čo sa ale snaží tento kód?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean ciska_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != 's') || (isbn.charAt(i+2) != 'b') || (isbn.charAt(i+3) != 'n')) {
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != 'S') || (isbn.charAt(i+2) != 'B') || (isbn.charAt(i+3) != 'N')) {
                return false;
            }
            if(isbn.charAt(i+4) == '-' && isbn.charAt(i+5) == '1' && isbn.charAt(i+6) == '3'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x') ||
(isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                ciska_zacali = true;
            }
        }
        if(ciska_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-'
&& isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```

Kód je ťažko zrozumiteľný

... asi sa snaží preskočiť ISBN prefix

# O čo sa ale snaží tento kód?

```
public static boolean isbn(String isbn){
    System.out.println(isbn);
    String fixed = "";
    boolean ciska_zacali = false;
    for(int i = 0; i < isbn.length(); i++){
        if(isbn.charAt(i) == 'i'){
            if ((isbn.charAt(i+1) != '3')){
                return false;
            }
        }
        if(isbn.charAt(i) == 'I'){
            if ((isbn.charAt(i+1) != '3')){
                return false;
            }
            if(isbn.charAt(i+4) == '.'){
                i = i+6;
                continue;
            }
        }
        if((isbn.charAt(i) >= 48 && isbn.charAt(i) <= 57) || (isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X')){
            if(isbn.charAt(i) == 'x' || isbn.charAt(i) == 'X'){
                fixed += 'x';
            } else {
                fixed += isbn.charAt(i);
                ciska_zacali = true;
            }
        }
        if(ciska_zacali && (isbn.charAt(i) < 48 || isbn.charAt(i) > 57) && isbn.charAt(i) != '-' && isbn.charAt(i) != ' ' && isbn.charAt(i) != 'x' && isbn.charAt(i) != 'X'){
            return false;
        }
    }
}
```

Kód je ťažko zrozumiteľný

```
int startIndex = 0;
if (isbn.startsWith("ISBN-13")) {
    startIndex += 7;
} else if (isbn.startsWith("ISBN")) {
    startIndex += 4;
} else if (isbn.startsWith("isbn")) {
    startIndex += 4;
}

int length = isbn.length();
for(int i = startIndex; i < length; ++i){
    char ch = isbn.charAt(i);
    ...
}
```

Síce menej efektívne, ale čitateľnejšie

**Gauss**

# Čo sa dá zlepšiť tu?

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
  
    //  $n = (b - a) / \text{delta} + 1$   
    BigInteger n = bb.subtract(aa).divide(d).add(one);  
    //  $x = (n - 1)n / 2 * \text{delta}$   
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    //  $n * a + x$   
    return n.multiply(aa).add(x).longValue();  
}
```

# Čo sa dá zlepšiť tu?

## Konštanty

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
  
    //  $n = (b - a) / \text{delta} + 1$   
    BigInteger n = bb.subtract(aa).divide(d).add(one);  
    //  $x = (n - 1)n / 2 * \text{delta}$   
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    //  $n * a + x$   
    return n.multiply(aa).add(x).longValue();  
}
```



# Čo sa dá zlepšiť tu?

## Konštanty

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
  
    //  $n = (b - a) / \text{delta}$   
    BigInteger n = bb.subtract(aa).divide(d);  
    //  $x = (n - 1)n / 2 * \text{delta}$   
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    //  $n * a + x$   
    return n.multiply(aa).add(x).longValue();  
}
```

# Čo sa dá zlepšiť tu?

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
  
    //  $n = (b - a) / \text{delta} + 1$   
    BigInteger n = bb.subtract(aa).divide(d).add(one);  
    //  $x = (n - 1)n / 2 * \text{delta}$   
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    //  $n * a + x$   
    return n.multiply(aa).add(x).longValue();  
}
```

# Čo sa dá zlepšiť tu?

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
  
    // n = (b - a) / delta + 1  
    BigInteger n = bb.subtract(aa).divide(d).add(one);  
    // x = (n - 1)n / 2 * delta  
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    // n * a + x  
    return n.multiply(aa).add(x).longValue();  
}
```

názvy premenných

# Čo sa dá zlepšiť tu?

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
  
    // n = (b - a) / delta + 1  
    BigInteger n = bb.subtract(bb).divide(d).add(one);  
    // x = (n - 1)n / 2 * d  
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    // n * a + x  
    return n.multiply(aa).add(x).longValue();  
}
```

názvy premenných

# Čo sa dá zlepšiť tu?

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
  
    //  $n = (b - a) / \text{delta} + 1$   
    BigInteger n = bb.subtract(aa).divide(d).add(one);  
    //  $x = (n - 1)n / 2 * \text{delta}$   
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    //  $n * a + x$   
    return n.multiply(aa).add(x).longValue();  
}
```

# Čo sa dá zlepšiť tu?

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);
```

Prečo ale takýto vzorec?

```
    // n = (b - a) / delta + 1  
    BigInteger n = bb.subtract(aa).divide(d).add(one);  
    // x = (n - 1)n / 2 * delta  
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    // n * a + x  
    return n.multiply(aa).add(x).longValue();  
}
```

# Čo sa dá zlepšiť tu?

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
  
    // n = (b - a) / delta + 1  
    BigInteger n = bb.subtract(aa).divide(d).add(one);  
    // x = (n - 1)n / 2 * delta  
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    // n * a + x  
    return n.multiply(aa).add(x).longValue();  
}
```

Prečo ale takýto vzorec?

(prvy + posledny) \* pocet / 2

# Čo sa dá zlepšiť tu?

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);
```

Prečo ale takýto vzorec?

```
    // n = (b - a) / delta + 1  
    BigInteger n = bb.subtract(aa).divide(d).add(one);  
    // x = (n - 1)n / 2 * delta  
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    // n * a + x  
    return n.multiply(aa).add(x).longValue();  
}
```

```
BigInteger first = bigA;  
BigInteger last = bigA.plus(n.minus(ONE).multiply(bigDelta));  
BigInteger sum = first.plus(last).multiply(n).divide(TWO);
```



# Čo sa dá zlepšiť tu?

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
  
    //  $n = (b - a) / \text{delta} + 1$   
    BigInteger n = bb.subtract(aa).divide(d).add(one);  
    //  $x = (n - 1)n / 2 * \text{delta}$   
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    //  $n * a + x$   
    return n.multiply(aa).add(x).longValue();  
}
```

# Čo sa dá zlepšiť tu?

**BigInteger je veľký kanón, zmestíme sa do long**

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
  
    //  $n = (b - a) / \text{delta} + 1$   
    BigInteger n = bb.subtract(aa).divide(d).add(one);  
    //  $x = (n - 1)n / 2 * \text{delta}$   
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    //  $n * a + x$   
    return n.multiply(aa).add(x).longValue();  
}
```

# Čo sa dá zlepšiť tu?

**BigInteger je veľký kanón, zmestíme sa do long**

```
public static long sucet(int a, int b, int delta) {  
    BigInteger one = BigInteger.valueOf(1), two = BigInteger.valueOf(2);  
    BigInteger aa = BigInteger.valueOf(a), bb = BigInteger.valueOf(b), d =  
    BigInteger.valueOf(delta);  
    int n = (b - a) / delta + 1;  
    // int first = a;  
    BigInteger last = first + (n-1)*delta;  
    // long sum = ((long)first + last) * n / 2;  
    BigInteger x = n.subtract(one).multiply(n).divide(two).multiply(d);  
    // n * a + x  
    return n.multiply(aa).add(x).longValue();  
}
```

**Dve ukážky nakoniec**

# Hovorí to samo za seba

```
public class Sedemciferne {
    public static int najvacsie7() {
        boolean parne = false;
        boolean trojkou = false;
        boolean stvorkou = false;
        boolean patkou = false;
        boolean setkou = false;
        boolean sedmickou = false;
        boolean osmickou = false;
        boolean devinou = false;
        int dvojky = 0;
        int trojky = 0;
        int stvorky = 0;
        int patky = 0;
        int sestky = 0;
        int sedmicky = 0;
        int osmicky = 0;
        int deviny = 0;
        for (int i = 9999999 ; i > 0; i--){
            int y = i;
            int num;
            while (y > 0){
                num = y % 10;
                y = y / 10;
                if (num == 2){
                    dvojky += 1;
                }
                if (num == 3){
                    trojky += 1;
                }
                if (num == 4){
                    stvorky += 1;
                }
                if (num == 5){
                    patky += 1;
                }
                if (num == 6){
                    sestky += 1;
                }
                if (num == 7){
                    sedmicky += 1;
                }
                if (num == 8){
                    osmicky += 1;
                }
                if (num == 9){
                    deviny += 1;
                }
                num = 0;
            }
            if ((dvojky > 1) || (trojky > 1) || (stvorky > 1) || (patky > 1) || (sestky > 1) || (sedmicky > 1) || (osmicky > 1) || (deviny > 1)){
                parne = false;
                trojkou = false;
                stvorkou = false;
                patkou = false;
                setkou = false;
                osmickou = false;
                devinou = false;
                dvojky = 0;
                trojky = 0;
                stvorky = 0;
                patky = 0;
                sestky = 0;
                sedmicky = 0;
                osmicky = 0;
                deviny = 0;
            }
            else {
                if (dvojky == 1) {
                    if ((i % 2) == 0) {
                        parne = true;
                    } else {
                        parne = false;
                    }
                }
                else {
                    parne = true;
                }
            }
            if (trojky == 1) {
                if ((i % 3) == 0){
                    trojkou = true;
                }
                else {
                    trojkou = false;
                }
            }
            else {
                trojkou = true;
            }
            if (stvorky == 1) {
                if ((i % 4) == 0) {
                    stvorkou = true;
                } else {
                    stvorkou = false;
                }
            }
            else {
                stvorkou = true;
            }
            if (patky == 1) {
                if ((i % 5) == 0) {
                    patkou = true;
                } else {
                    patkou = false;
                }
            }
            else {
                patkou = true;
            }
            if (sestky == 1) {
                if ((parne == true) & (trojkou == true)) {
                    setkou = true;
                } else {
                    setkou = false;
                }
            }
        }
    }
}
```

# Hovorí to samo za seba

```
public static int najvacsie7() {
    for(int a=9;a>0;a--){
        for(int b=9;b>0;b--){
            if(b==a) continue;
            for(int c=9;c>0;c--){
                if(c==b || c==a) continue;
                for(int d=9;d>0;d--){
                    if(d==c || d==b || d==a) continue;
                    for(int e=9;e>0;e--){
                        if(e==d || e==c || e==b || e==a) continue;
                        for(int f=9;f>0;f--){
                            if(f==e || f==d || f==c || f==b || f==a) continue;
                            for(int g=9;g>0;g--){
                                if(g==f || g==e || g==d || g==c || g==b || g==a) continue;
                                if(a==2 || b==2 || c==2 || d==2 || e==2 || f==2 || g==2){
                                    if(g%2!=0) continue;
                                }
                                if(a==3 || b==3 || c==3 || d==3 || e==3 || f==3 || g==3){
                                    if((a+b+c+d+e+f+g)%3!=0) continue;
                                }
                                if(a==4 || b==4 || c==4 || d==4 || e==4 || f==4 || g==4){
                                    if((f*10+g)%4!=0) continue;
                                }
                                if(a==5 || b==5 || c==5 || d==5 || e==5 || f==5 || g==5){
                                    if(g%5!=0) continue;
                                }
                                if(a==6 || b==6 || c==6 || d==6 || e==6 || f==6 || g==6){
                                    if(g%2!=0 || (a+b+c+d+e+f+g)%3!=0) continue;
                                }
                                if(a==7 || b==7 || c==7 || d==7 || e==7 || f==7 || g==7){
                                    if((a*1000000+b*100000+c*10000+d*1000+e*100+f*10+g)%7!=0) continue;
                                }
                                if(a==8 || b==8 || c==8 || d==8 || e==8 || f==8 || g==8){
                                    if((e*100+f*10+g)%8!=0) continue;
                                }
                                if(a==9 || b==9 || c==9 || d==9 || e==9 || f==9 || g==9){
                                    if((a+b+c+d+e+f+g)%9!=0) continue;
                                }
                                return a*1000000+b*100000+c*10000+d*1000+e*100+f*10+g;
                            }
                        }
                    }
                }
            }
        }
    }
    return -1;
}
```