Ćwiczenia z programowania obiektowego
20.04.2011 r.

URI (ang. *Uniform Resource Identifier*) jest standardem internetowym umożliwiającym łatwą identyfikację zasobów w sieci. Poniższy kod prezentuje szkielet walidatora sprawdzającego poprawność URI.

```
/**
 *  An example URI and its component parts. [rfc3986]
 *
 *  foo://example.com:8042/over/there?name=ferret#nose
 *  \_/   \______________/\_________/ \_________/ \__/
 *   |           |            |            |        |
 * scheme     authority       path        query   fragment
 *
 * The pattern is
 * <scheme> : <authority> / <path> ? <query> # <fragment>
 */

public class CustomURIValidator {
    String scheme, authority, path, query, fragment;

    public CustomURIValidator(String uri) throws ...  {
        ...
        validateScheme();
        validateAuthority();
        validatePath();
    }
    /** @return true if c is either ALPHA, DIGIT, '-', '.', '_' or '~' **/
    public static boolean isUnreservedCharacter(char c) { ... }

    private void validateAuthority() throws EmptyComponentException,
                                            ComponentTooLongException {
        if (authority == null || authority.length() < 1)
            throw new EmptyComponentException("authority");
        if (authority.length() > 255)
            throw new ComponentTooLongException("authority",255);
    }

    private void validatePath() throws ComponentTooLongException,
                                            IllegalCharacterException {
        //leave blank
    }

    private void validateScheme() throws ... { ... }
}
```

1. Uzupełnij konstruktor klasy `CustomURIValidator` o kod dzielący URI na składowe i przypisujący składowe do odpowiednich pól walidatora.

2. Uzupełnij metodę `validateScheme()` tak, aby spełnione były następujące wymagania:

   - schemat musi być niepusty.
   - schemat nie może być dłuższy niż 255 znaków.
   - schemat może zawierać jedynie litery, cyfry, myślnik, kropkę, znak podkreślenia lub tyldę (tzw. *unreserved characters*).

   Jeśli któreś z powyższych wymagań nie jest spełnione, powinien być rzucony odpowiedni wyjątek.

```
public class IllegalCharacterException extends Exception {
    public IllegalCharacterException(String msg) { super(msg); }
}

public class EmptyComponentException extends Exception {
    public EmptyComponentException(String msg) { super(msg); }
}

public class ComponentTooLongException extends Exception {
    public ComponentTooLongException(String component, int expected) {
            super("The "+component+ " component can not be longer then"
                                        +expected);
    }
}
```

3. Dodajmy nowy wyjątek `InvalidCustomURIException`:

```
public class InvalidCustomURIException extends Exception {/*...*/}
public class IllegalCharacterException extends InvalidCustomURIException {/*...*/}
public class EmptyComponentException extends InvalidCustomURIException {/*...*/}
public class ComponentTooLongException extends InvalidCustomURIException {/*...*/}
```

   Zmień kod konstruktora klasy `CustomURIValidator` w ten sposób, aby rzucany był jeden wyjątek `InvalidCustomURIException` zawierający w sobie informacje o wszystkich błędach ze wszystkich składowych. Możesz zmienić kod wyjątków.