***Builder Pattern***

Gamma Categorization: Creational

Example:

String [] words = {"hello", "world"};

StringBuilder sb = new StringBuilder();

sb.**append**("<ul>\n");

for (String word: words){

sb.**append**(**String.format**(" <li>**%s**</li>**\n**", word)); // zwróci Stringi: <li>hello</li> oraz <li>world<\li>

}

sb.**append**("</ul>");

Example:

private final String newLine = **System.lineSeparator();**

Example:

private String toStringImpl(int indent)

{

StringBuilder sb = new StringBuilder();

String space = **String.join**("", **Collections.nCopies**(indent \* indentSize, " "));

**sb.append(String.format("%s<%s>%s", space, name, newLine));** // zwróci String z wartości

if (text != null && !text.isEmpty())

{

sb.append(String.join("", Collections.nCopies(indentSize\*(indent+1), " ")))

.append(text)

.append(newLine);

}

for (HtmlElement e : elements)

sb.append(e.toStringImpl(indent + 1));

sb.append(String.format("%s</%s>%s", space, name, newLine));

return sb.toString();

}

**Collections.nCopies[intValue, object]** – zwraca tabelę elemenetów object w liczbie intValue (wielkość tablicy wynosi intValue)

**String.join(separator, array)** – zwraca Stringa z wartości w tablicy array oddzielonych separatorem np. separator = "" (pusty String)

Example:

class PersonBuilder **<T extends PersonBuilder<T>>** {

protected Person person = new Person();

// critical to return T here

public T withName(String name) {

person.name = name;

return getMyInstance ();

}

protected T getMyInstance () {

// unchecked cast, but actually safe

// proof: try sticking a non-PersonBuilder as T parameter; it won't work!

**return (T) this**; // zwraca instancję EmployeeBuilder’a

}

public Person build() {

return person;

}

}

class EmployeeBuilder **extends PersonBuilder<EmployeeBuilder>** {

public EmployeeBuilder worksAs(String posiation) {

person.position = position;a

return getMyInstance ();

}

@Override

protected EmployeeBuilder getMyInstance () {

**return this;** // zwraca instancję EmployeeBuilder’a

}

}

Dzięki temu możemy korzystać z fluent builder’a:

EmployeeBuilder eb = new EmployeeBuilder()

.withName("Dmitri")

.worksAs("Quantitative Analyst");