МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний Технічний Університет України
«Київський Політехнічний Інститут»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №9
з дисципліни «Інженерія програмного забезпечення»
на тему: «Породжувальні шаблони. Шаблони Abstract Factory, Builder»

Виконав:
студент 2-го курсу ФІОТ
групи ІВ-71
Мазан Я.В.
Номер залікової книжки: 7109
Варіант: 3

Перевірив:
доцент кафедри ОТ
Антонюк А.І.

Київ – 2018

Завдання

Варіант – 3

Визначити специфікації класів для прямокутного ігрового простору та завантажувача його конфігурації із зовнішнього файлу.

Код програми

Main.java:

```
package com.lab111.labwork9;
/**
 * Main class
 * @author - Yan Mazan
 * @version - 1.0
 */
public class Main {
    /**
     * program test
     * creates a Gamespace with builder, then saves it to "file"
     * creates a GamespaceLoader from GamespaceLoaderBuilder which then recreates the
Gamespace from the "file"
     * @param args - command line arguments
     */
    public static void main(String[] args) {
        int height = 100;
        int width = 200;
        String biome = "tundra";
        int level = 22;
        int enemies_num = 88;
        int[] start_pos = {40,22};
        GamespaceBuilder builder = new GamespaceBuilder();
        builder.setHeight(height);
        builder.setWidth(width);
        builder.setBiome(biome);
        builder.setLevel(level);
        builder.setEnemiesNum(enemies_num);
        builder.setStartPos(start_pos[0],start_pos[1]);
        Gamespace gamespace = builder.result();
        String[] file = gamespace.save_to_file("Game 1");
        GamespaceLoaderBuilder loader_builder = new GamespaceLoaderBuilder();
        GamespaceLoader loader = loader_builder.result();
        loader.analyse_file(file);
        Gamespace recreated_gamespace = loader.result();
        String[] file2 = recreated_gamespace.save_to_file("Game 2");
        System.out.println("Gamespace 1 raw data: " + file[1]);
        System.out.println("\nSaving gamespace 1 to file \"Game 1\"...\nCreating
gamespace 2 from Game 1...\nSaving gamespace 2 to file \"Game 2\"...\n");
        System.out.println("Gamespace 2 raw data: " + file2[1]);
    }
}
```

Builder.java:

```
package com.lab111.labwork9;
/**
 * template of Builder for classes Gamespace and GamespaceLoader
 * @author - Yan Mazan
 * @version - 1.0
 */
public interface Builder {
    void reset();
```

```
    void setHeight(int height);
    void setWidth(int width);
    void setBiome(String name);
    void setLevel(int level);
    void setEnemiesNum(int num);
    void setStartPos(int x, int y);
}
```

Gamespace.java:

```
package com.lab111.labwork9;
/**
 * Gamespace class
 * @author - Yan Mazan
 * @version - 1.0
 */
public class Gamespace {
    private final int height;
    private final int width;
    private final String biome;
    private final int level;
    private final int enemies_num;
    private final int[] start_pos;
    /**
     * Constructor of a gamespace with all parameters defined
     * @param height - gamespace height
     * @param width - gamespace width
     * @param biome - gamespace biome
     * @param level - gamespace game level
     * @param enemies_num - enemies number in gamespace
     * @param x - x coord of starting position
     * @param y - y coord of starting position
     */
    public Gamespace(int height, int width, String biome, int level, int enemies_num,
int x, int y) {
        this.height = height;
        this.width = width;
        this.biome = biome;
        this.level = level;
        this.enemies_num = enemies_num;
        this.start_pos = new int[]{x,y};
    }
    /**
     * Saving of gamespace into String representation of file
     * @param name - name of a "file" to save in
     * @return - tuple with the first element being a name of a "file"
     *           and raw data of a gamespace in the second element
     */
    public String[] save_to_file(String name) {
        String file_content = (new StringBuilder("" + height + "\n"
                + width + "\n"
                + biome + "\n"
                + level + "\n"
                + enemies_num + "\n"
                + start_pos[0] + "\n"
                + start_pos[1])).toString();
        return new String[]{name,file_content};
    }
}
```

GamespaceLoader.java:

```
package com.lab111.labwork9;
/**
 * GamespaceLoader class that loads Gamespace from a "file"
```

```
 * @author - Yan Mazan
 * @version - 1.0
 */
public class GamespaceLoader {
    private int height;
    private int width;
    private String biome;
    private int level;
    private int enemies_num;
    private int[] start_pos;
    /**
     * Constructor of a gamespace loader with all parameters defined
     * @param height - gamespace height
     * @param width - gamespace width
     * @param biome - gamespace biome
     * @param level - gamespace game level
     * @param enemies_num - enemies number in gamespace
     * @param x - x coord of starting position
     * @param y - y coord of starting position
     */
    public GamespaceLoader(int height, int width, String biome, int level, int
enemies_num, int x, int y) {
        this.height = height;
        this.width = width;
        this.biome = biome;
        this.level = level;
        this.enemies_num = enemies_num;
        this.start_pos = new int[]{x,y};
    }
    /**
     * method that analyses a file and changes output Gamespace class attributes values
     * @param file - file to analyse
     */
    public void analyse_file(String[] file) {
        String[] raw_data = file[1].split("\n");
        this.height = Integer.parseInt(raw_data[0]);
        this.width = Integer.parseInt(raw_data[1]);
        this.biome = raw_data[2];
        this.level = Integer.parseInt(raw_data[3]);
        this.enemies_num = Integer.parseInt(raw_data[4]);
        this.start_pos = new int[]
{Integer.parseInt(raw_data[5]),Integer.parseInt(raw_data[6])};
    }
    /**
     * Method to output result Gamespace after "file" analysing
     * @return - new Gamespace object
     */
    public Gamespace result() {
        GamespaceBuilder res = new GamespaceBuilder();
        res.setHeight(height);
        res.setWidth(width);
        res.setBiome(biome);
        res.setLevel(level);
        res.setEnemiesNum(enemies_num);
        res.setStartPos(start_pos[0],start_pos[1]);
        return res.result();
    }
}
```

GamespaceBuilder.java:

```
package com.lab111.labwork9;
/**
 * Builder class for Gamespace class
 * @author - Yan Mazan
```

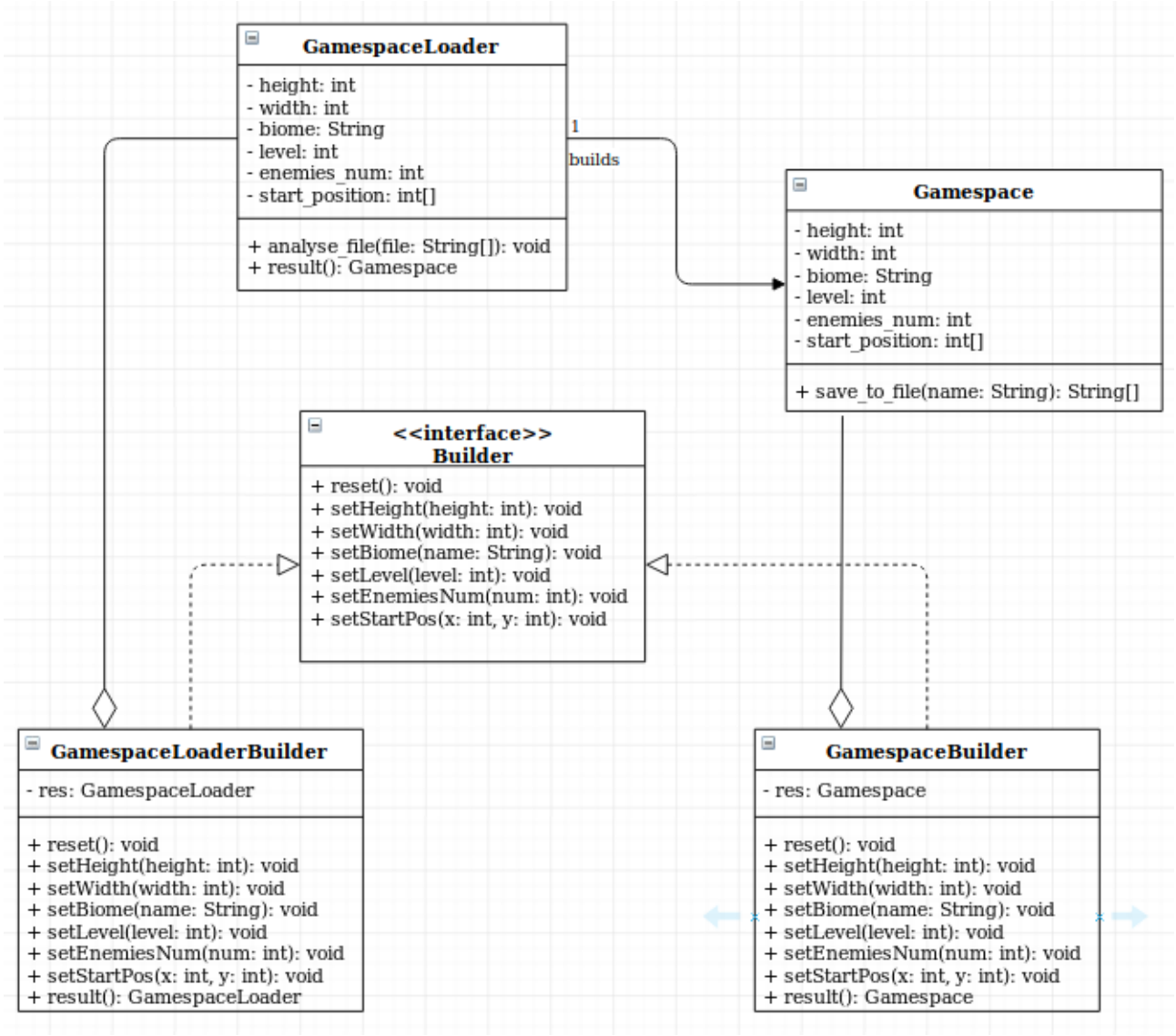
```
 * @version - 1.0
 */
public class GamespaceBuilder implements Builder {
    private Gamespace res;
    private int height;
    private int width;
    private String biome;
    private int level;
    private int enemies_num;
    private int[] start_pos;
    /**
     * Method that resets attributes of output Gamespace object
     */
    public void reset() {
        height = 0;
        width = 0;
        level = 0;
        enemies_num = 0;
        start_pos = new int[]{0,0};
        biome = null;
        res = new
Gamespace(height,width,biome,level,enemies_num,start_pos[0],start_pos[1]);
    }
    /**
     * Methods that set gamespace parameters independently
     */
    public void setHeight(int height) {
        this.height = height;
    }
    public void setWidth(int width) {
        this.width = width;
    }
    public void setBiome(String name) {
        this.biome = name;
    }
    public void setLevel(int level) {
        this.level = level;
    }
    public void setEnemiesNum(int num) {
        this.enemies_num = num;
    }
    public void setStartPos(int x, int y) {
        this.start_pos = new int[]{x,y};
    }
    /**
     * Method to output a new Gamespace object with preset attributes
     * @return - Gamespace object
     */
    public Gamespace result() {
        res = new
Gamespace(height,width,biome,level,enemies_num,start_pos[0],start_pos[1]);
        return res;
    }
}
```

GamespaceLoaderBuilder.java:

```
package com.lab111.labwork9;
/**
 * Builder class for GamespaceLoader class
 * @author - Yan Mazan
 * @version - 1.0
 */
public class GamespaceLoaderBuilder implements Builder {
    private GamespaceLoader res;
```

```
    private int height;
    private int width;
    private String biome;
    private int level;
    private int enemies_num;
    private int[] start_pos;
    /**
     * Class constructor
     */
    public GamespaceLoaderBuilder() {
        reset();
    }
    /**
     * Method that resets attributes of output GamespaceLoader object
     */
    public void reset() {
        height = 0;
        width = 0;
        level = 0;
        enemies_num = 0;
        start_pos = new int[]{0,0};
        biome = null;
        res = new
GamespaceLoader(height,width,biome,level,enemies_num,start_pos[0],start_pos[1]);
    }
    /**
     * Methods that set gamespace parameters independently
     */
    public void setHeight(int height) {
        this.height = height;
    }
    public void setWidth(int width) {
        this.width = width;
    }
    public void setBiome(String name) {
        this.biome = name;
    }
    public void setLevel(int level) {
        this.level = level;
    }
    public void setEnemiesNum(int num) {
        this.enemies_num = num;
    }
    public void setStartPos(int x, int y) {
        this.start_pos = new int[]{x,y};
    }
    /**
     * Method to output a new GamespaceLoader object with preset attributes
     * @return - GamespaceLoader object
     */
    public GamespaceLoader result() {
        res = new
GamespaceLoader(height,width,biome,level,enemies_num,start_pos[0],start_pos[1]);
        return res;
    }
}
```

Діаграма класів



Тестування програми

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin
Gamespace 1 raw data: 100
200
tundra
22
88
40
22

Saving gamespace 1 to file "Game 1"...
Creating gamespace 2 from Game 1...
Saving gamespace 2 to file "Game 2"...

Gamespace 2 raw data: 100
200
tundra
22
88
40
22

Process finished with exit code 0
```

Висновок

Ознайомились із породжувальними шаблонами Abstract, Factory, Builder. Отримали навички із застосування шаблонів. Розроблена відповідна тестова програма. В даному випадку для розв’язання даної мені задачі довелось застосувати шаблон Builder для того, щоби не створювати об’єкти класів за допомогою великих конструкторів. Результати успішної роботи тестової програми наведені вище підтверджують правильність обраних рішень.