Федеральное государственное автономное образовательное учреждение

высшего образования

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа «Киберфизические системы и управление»

**Отчет №9**

по дисциплине «Системный подход к разработке программного обеспечения»

|  |  |  |
| --- | --- | --- |
| Выполнил:  студент гр. 3530902/00201 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Позолотин О. В. |
|  | <*подпись*> |  |
|  |  |  |
|  |  |  |
| Руководитель:  Кандидат т.н. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Нестеров С. А. |
|  | <*подпись*> |  |

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 г.

Санкт-Петербург

2022

**Ход работы**

**По уроку 7.4**

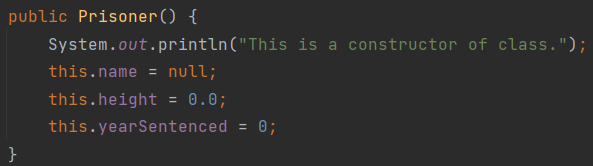


Рисунок 1 – конструктор без параметров

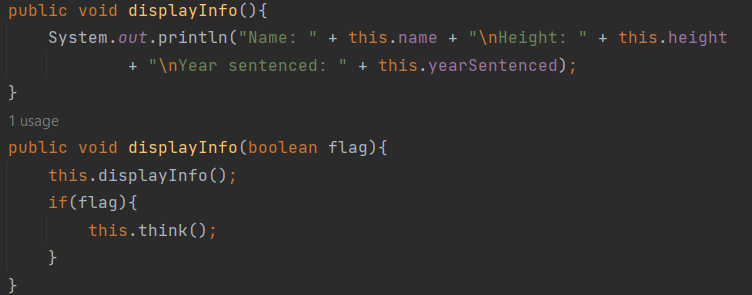


Рисунок 2 – два перегруженных метода, выводящие информацию о заключенном

Чтобы минимизировать повторения кода, мы вызываем в методе с параметром метод без параметров, выводящий информацию о заключенном.

**По уроку 7.5**

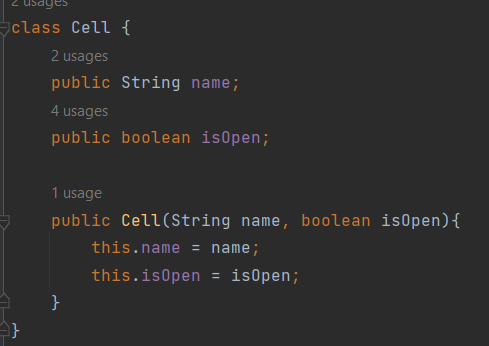
****

Рисунок 3 – класс Cell

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 – создаём поле типа Cell в классе Prisoner

Создаём поле и передаём в его конструктор параметры.

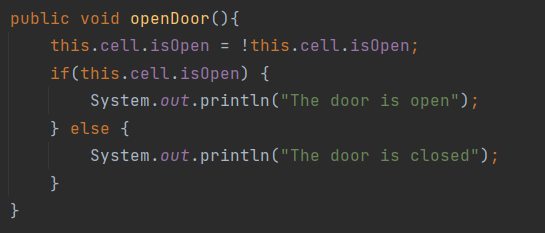


Рисунок 5 – метод openDoor в классе Prisoner

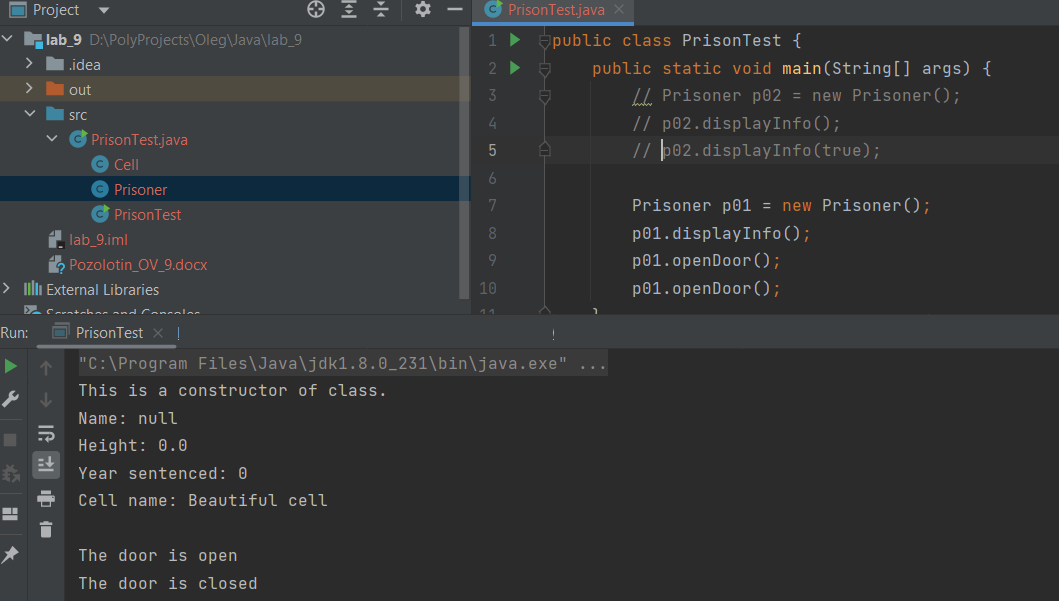


Рисунок 6 – результат вызова метода

Вызываем метод displayInfo один раз и два раза метод openDoor и видим, что состояние двери меняется (открывается и закрывается).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7 – делаем поля Cell private

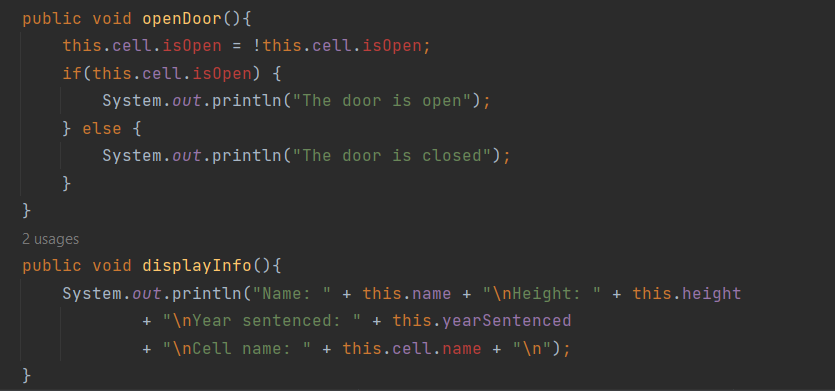


Рисунок 8 – проблемы

После того, как мы сделали поля Cell private у нас появились проблемы в методах openDoor и displayInfo класса Prisoner, так как мы уже не имеем доступа к полям класса по объекту.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 – модифицированный метод openDoor

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 – изменённый метод set для открытия/закрытия двери

Теперь у класса Cell есть private поле code и мы проверяем на его на корректность при попытке открытия двери.

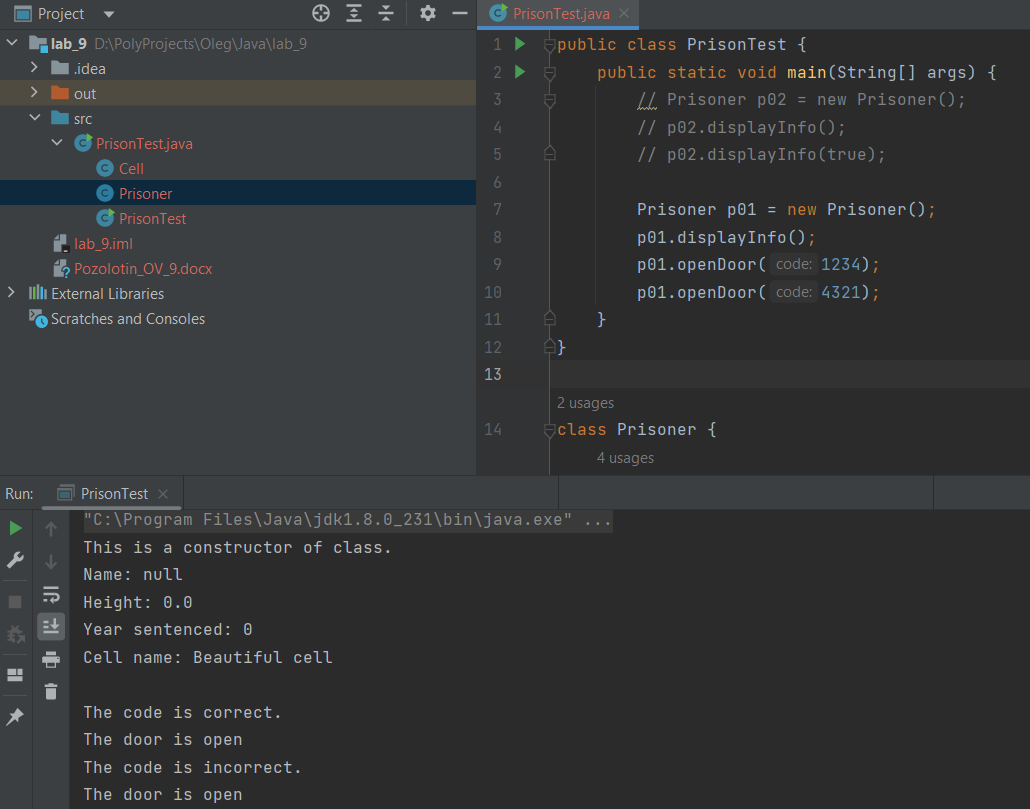


Рисунок 11 – результат работы программы

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 – действие через объект Cell

Теперь у нас нет метода openDoor для Prisoner’а, создаём объект типа Cell и проверяем открытость/закрытость двери через него.

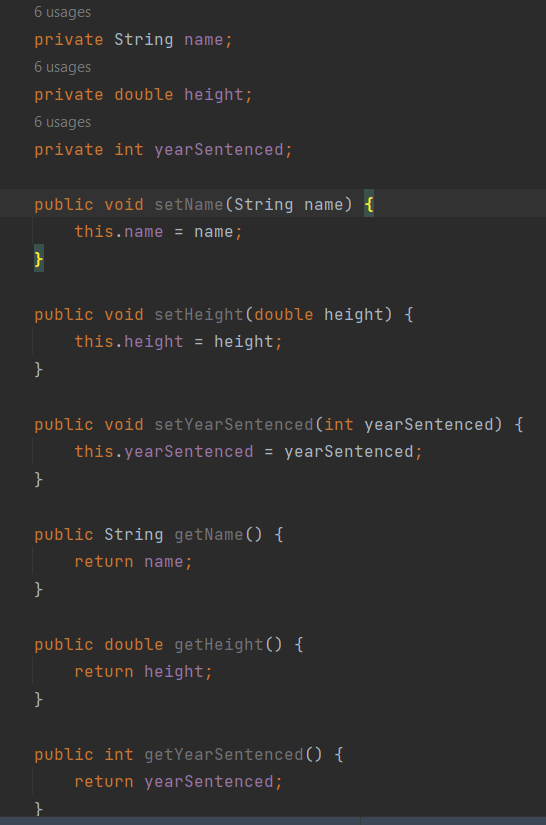


Рисунок 13 – сделали поля для Prisoner’а private и добавили для каждого поля getters и setters

**Ex\_1**

****

Задаём публичным полям значения и выводим их

**Ex\_2**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

Задаём значения полей двух объектов через setters. И далее выводим их.

**Ex\_3**

**Изображение выглядит как текст

Автоматически созданное описание**

В данном примере демонстрируется перегрузка двух методов (есть методы с параметрами и без параметров).

**Ex\_4**

Изображение выглядит как текст

Автоматически созданное описание

В данном примере для объектов одного класса используют перегруженные конструкторы (один без параметров, другой с параметрами).

**Ex\_5**

**Изображение выглядит как текст

Автоматически созданное описание**

В этом примере используются статические поля класса для подсчёта количества созданных объектов и выдаче каждому из них порядкового номера при создании.

**Ex\_6**

**Изображение выглядит как текст

Автоматически созданное описание**

В данном примере создаётся 3 объекта класса с именами и далее для второго объекта изменяется его имя через setter поля name.