Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

ОТЧЕТ по лабораторной работе №5

по дисциплине «Системный подход в разработке программного обеспечения»

Выполнил:

студент группы 5130902/20201 \_\_\_\_\_\_\_\_\_\_\_\_\_ А. И. Сафонов

подпись

**Проверил**:

Кандитад тех. Наук., доцент \_\_\_\_\_\_\_\_\_\_\_\_ С. А. Нестеров

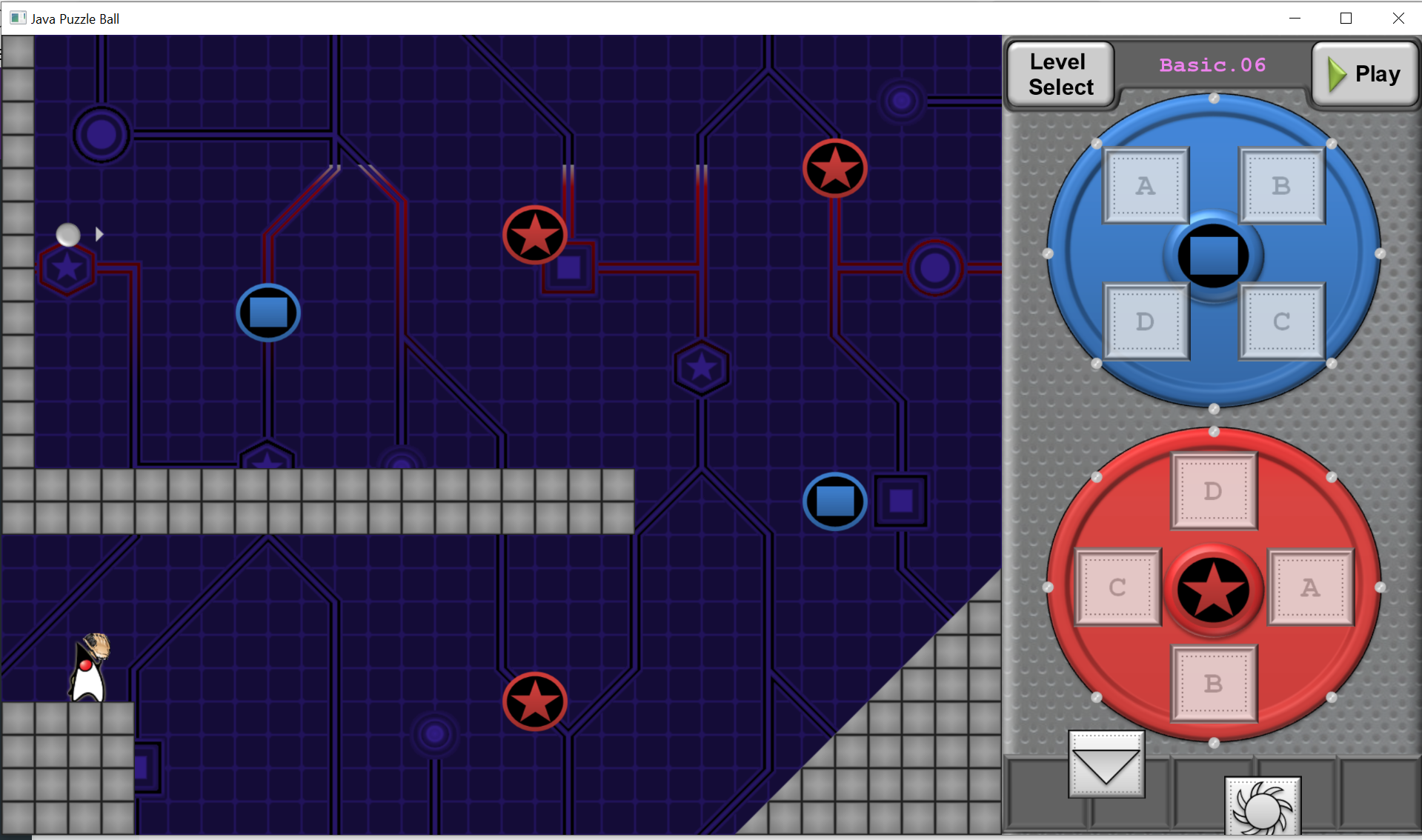
подпись

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024г.

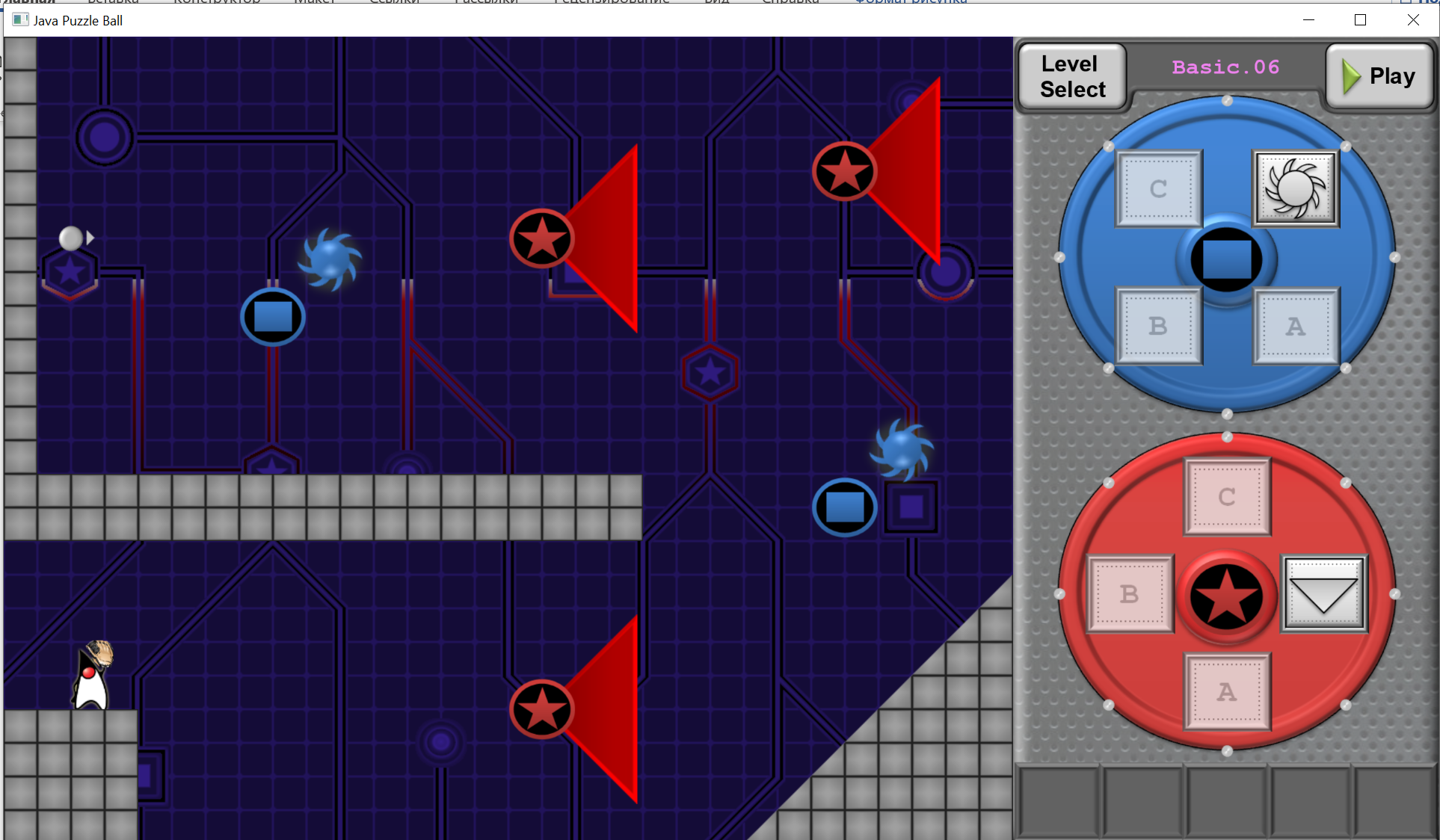
Санкт-Петербург, 2024

1. К занятию 7.1
   1. Упражнение 1

Нам необходимо довести мяч до дюка. Рекомендуется использовать уровень Basic.06 или выше. Выберем Basic.06. Базово уровень выглядит следующим образом:



Прохождение уровня



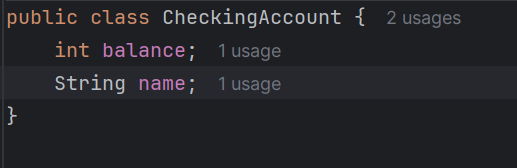
Для решения задания мы воспользовались инструментом, который отбивает мяч и инструментом, который превращает мяч в лезвие. Лезвие может разрушить одно препятствие, после чего мяч придет в изначальную форму.

Круги в правой части экрана можно назвать классами, а игровые значки в игровом поле – экземпляры этих классов.

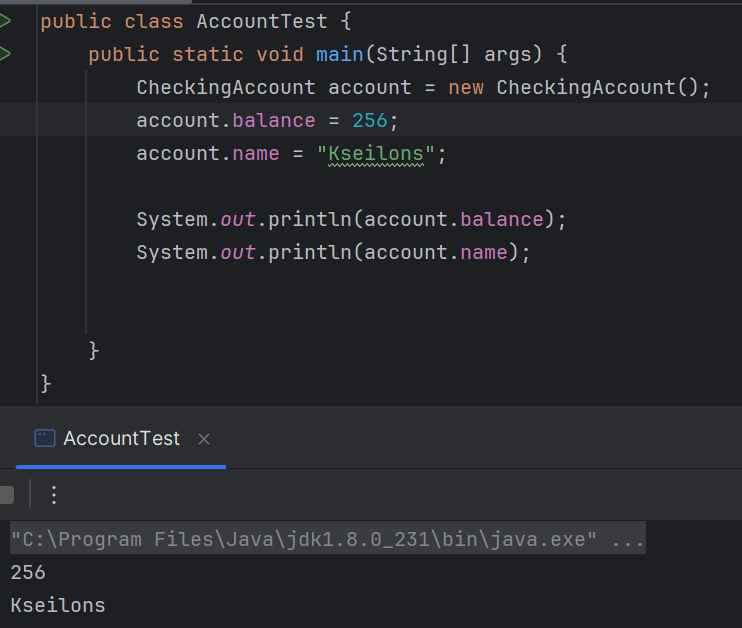
* 1. Упражнение 2

Сначала создадим новый проект. В нем создадим два класса. Создадим экземпляр одного класса в другом и назначим значения для полей первого.

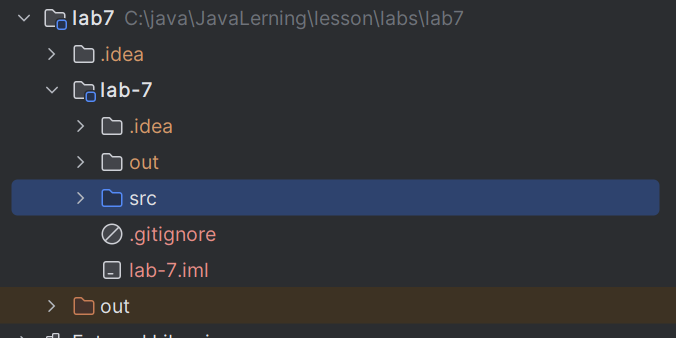
Класс CheckingAccount:



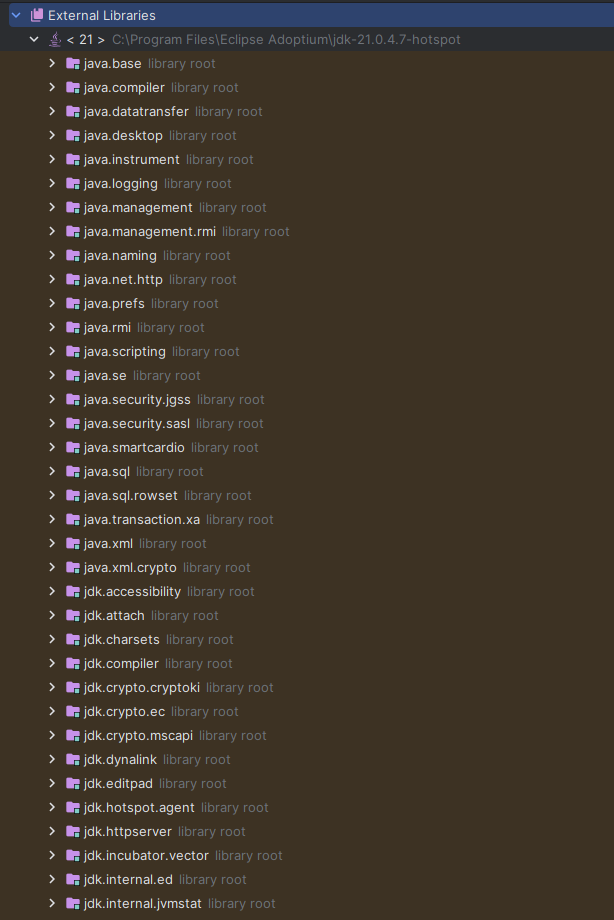
Класс AccountTest:



Теперь рассмотрим папку проекта.



Папка .idea и файл .iml содержат в себе файлы настроек проекта. В каталог out помещаются результаты компиляции программы. Этот каталог не является частью проекта, его можно очистить и ничего не поломается. Папка src хранит в себе исходный код. Gitignore позволяет не . Затем дан список с расширяющими библиотеками. Они выглядят следующим образом



* 1. Упражнение 3

Необходимо продолжить проект редактировать проект из прошлого задания

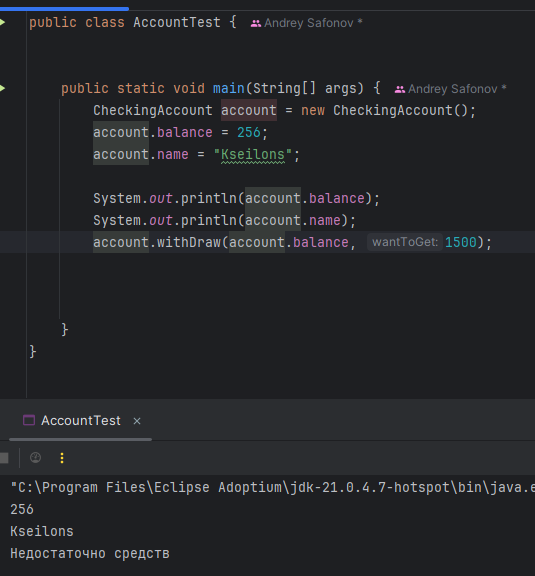
Необходимо реализовать метод, который проверяет счет по условию задания

Реализация этого метода

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Проверка работоспособности



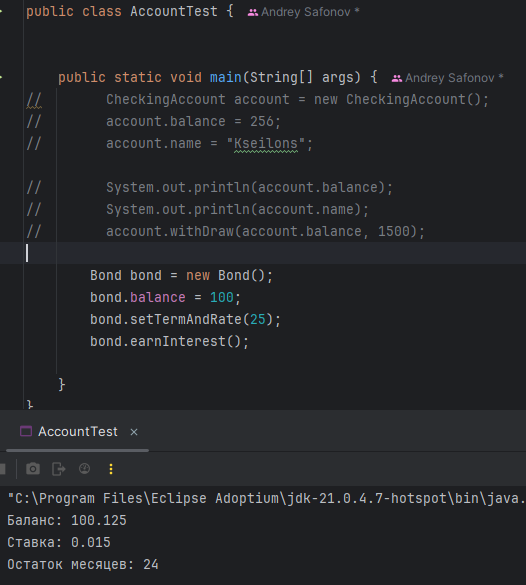
* 1. Упражнение 4

Нужно продолжить редактирование проекта, который используется в 2 и 3 упражнениях.

Создадим класс Bond, который отвечает требованиям задания

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

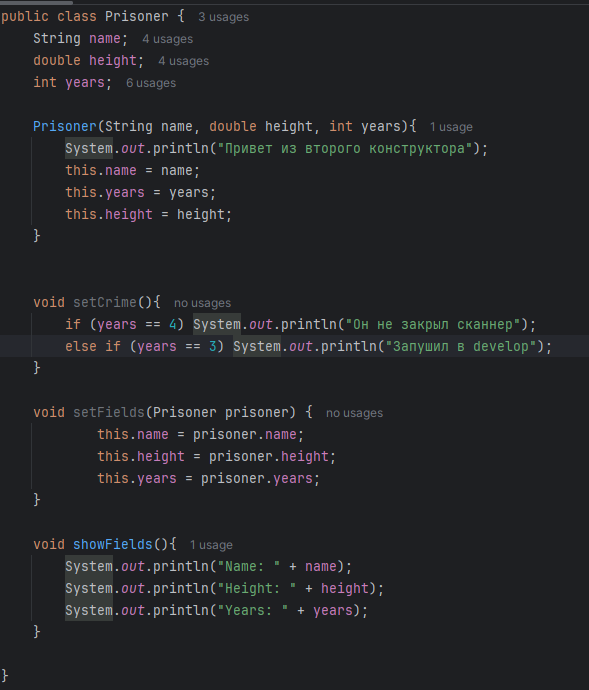
Автоматически созданное описание



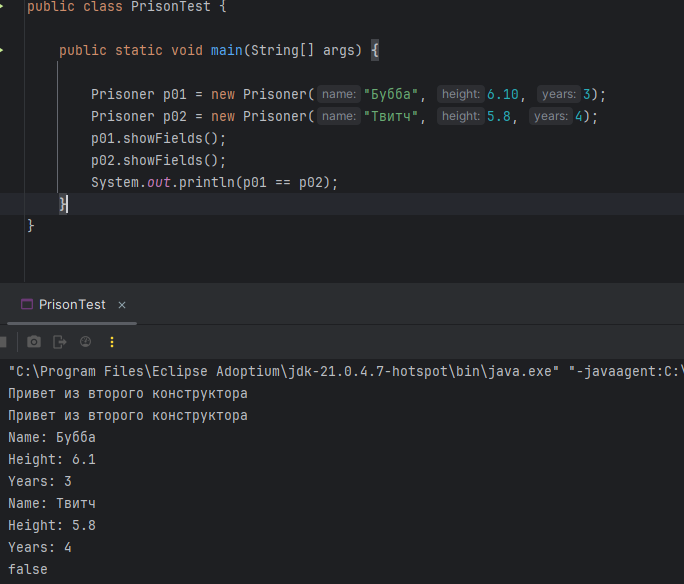
Пример работы программы

1. К занятию 7.2
   1. Упражнение 1

Создадим новый проект и 2 класса в нем. Класс преступника

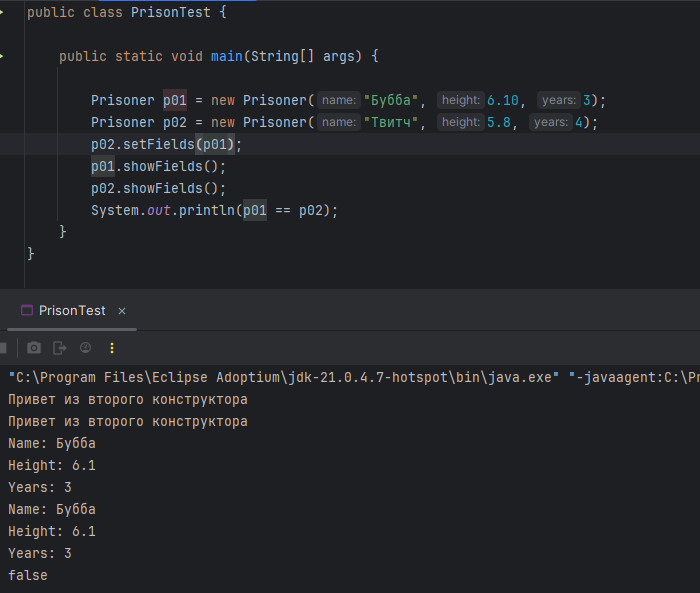


В тестовом классе смоделируем первую ситуацию и посмотрим, что выведется:

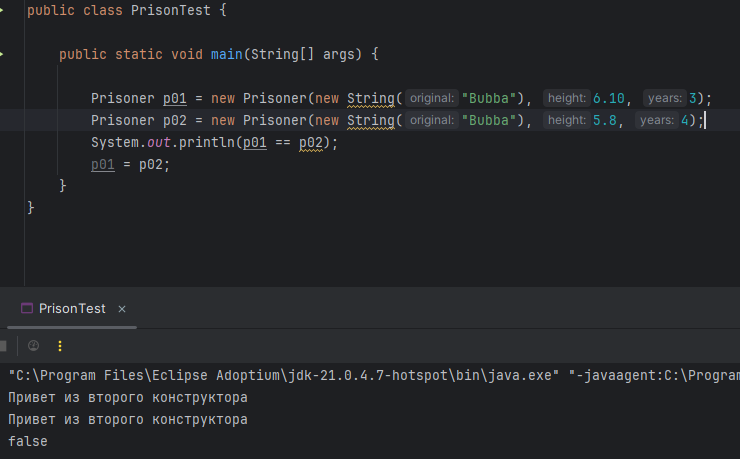


Оператор «==» сравнивает ссылки на объекты, а не значения этих объектов. В данном случае у объектов разные значения и разные адреса в памяти, поэтому подтвердить первое предложение нельзя.

Рассмотрим вторую ситуацию.



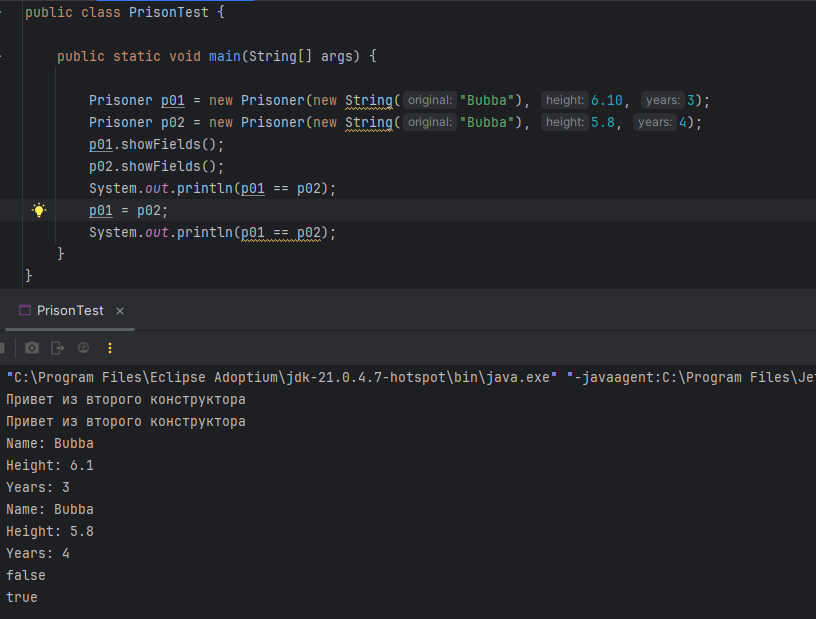
Видим, что значения у переменных равны. Потому можно сделать вывод, что оператор равно сравнивает ячейки памяти.



Действительно, ссылки разные.

* 1. Упражнение 2

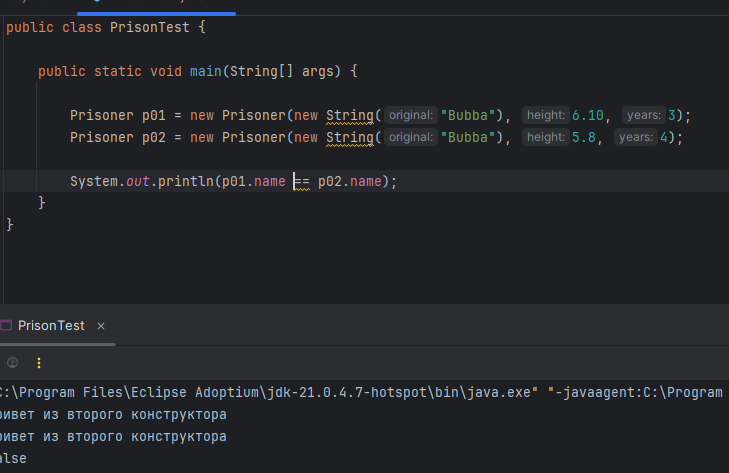
Работаем с проектом из предыдущего упражнения



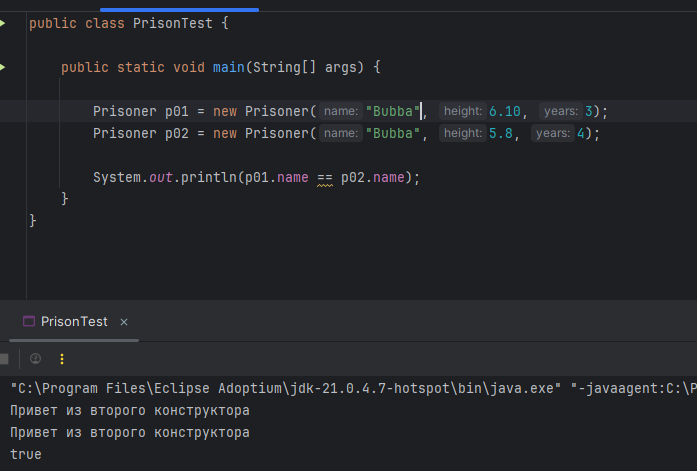
Видим, что если переменной bubba присвоим ссылку на twitch, то получим равенство.

* 1. Упражнение 3

Продолжаем работать с этим проектом.



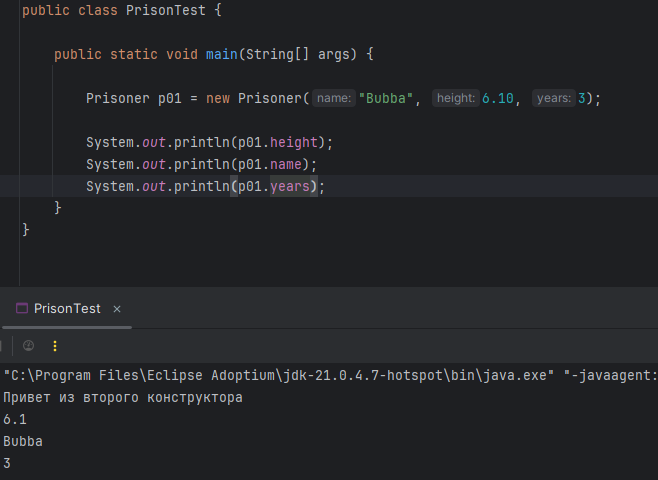
При создании строк с использованием ключевого слова new в пул строк в любом случае добавляется новая строка. Если же не использовать этот оператор, то будет проверяться, есть ли эта строка в пуле. Если да, то строка будет ссылаться на уже имеющийся адрес памяти, иначе в пул добавится новая строка.



Видим, что верхний вывод правильный.

1. Занятие 7.3
   1. Упражнение 1

Нам нужно работать с предыдущим проектом. Сначала узнаем, какие значения выведутся до инициализации полей.

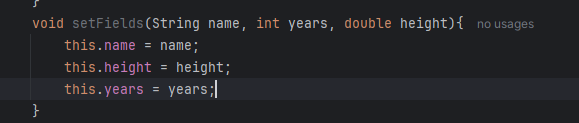


Видим, что выводятся значения по умолчанию.

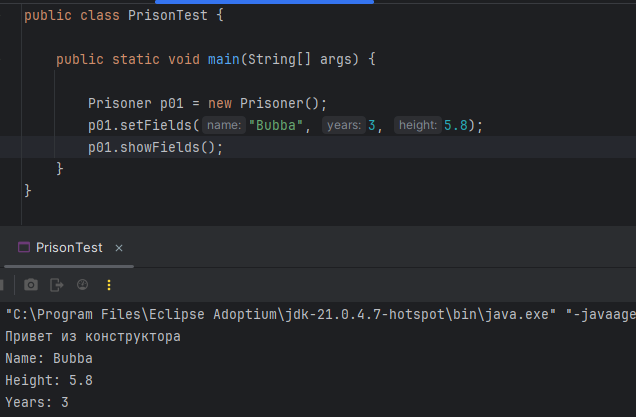
* 1. Упражнение 2

Нам необходимо добавить сеттер, который установит значение 3 полям.

Реализация сеттера

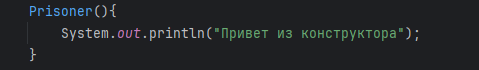


Работа программы

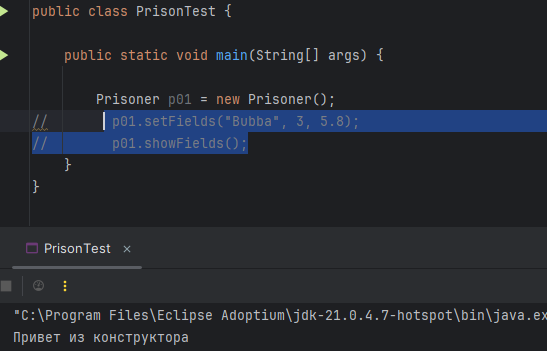


* 1. Упражнение 3

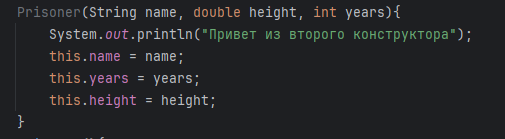
Продолжим редактировать проект. Нам нужно изменить конструктор.



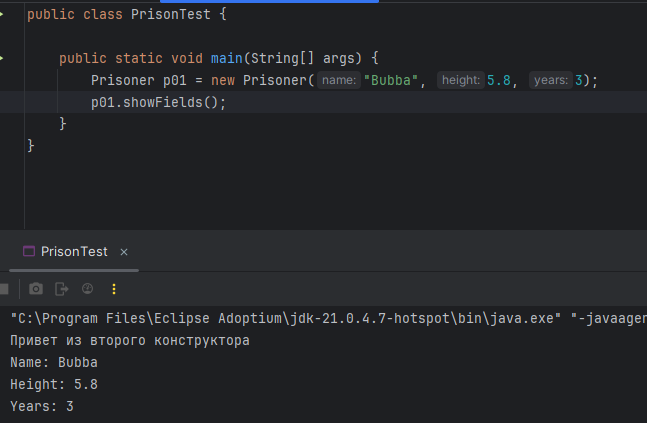
Что происходит при создании объекта



Далее нужно создать конструктор, который установит значение полей.



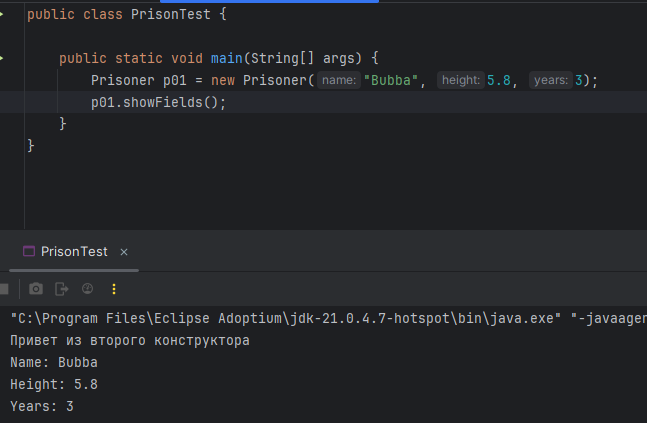
Результат выполнения программы



* 1. Упражнение 4

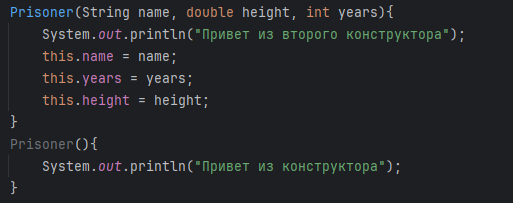
Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание



1. Занятие 7.4
   1. Упражнение 2

Нужно перегрузить конструктор, то есть он должен уметь принимать разные аргументы.



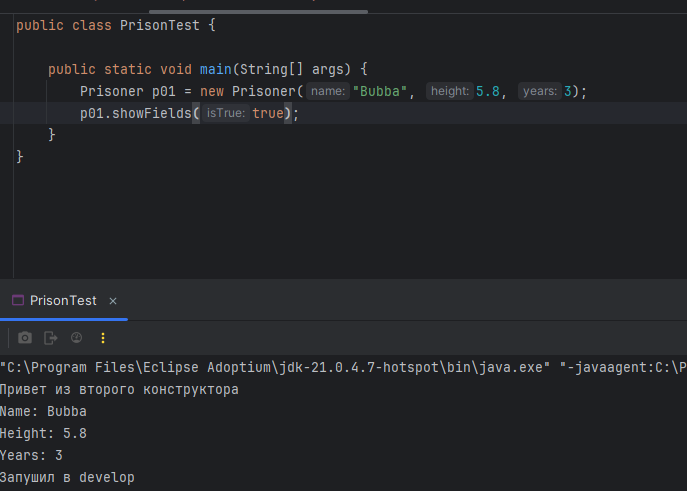
* 1. Упражнение 3

Нужно написать два метода по заданным условиям.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Результат работы



* 1. Упражнение 4



Конспект по книге

Автор сравнивает класс с чертежом детали, а объекты – с конкретными деталями. В описании класса содержится информация о том, какие поля и методы будут иметь создаваемые объекты. Поля описываются так же, как и переменные.

Создание объекта делится на два этапа:

1. Объявление объектной переменной, через которую получим доступ к объекту
2. Создание объекта и связывание с переменной.

Объявление переменной не говорит о создании объекта. Объект создается с помощью оператора new и конструктора. Переменная является ссылкой на объект.

Методы и конструкторы

Есть такое понятие, как перегрузка методов. Это значит, что может быть множество методов с одинаковым названием. При этом аргументы должны быть различными.

Конструкторы – методы, которые вызываются автоматически при создании объекта класса. Особенности конструктора:

1. Имя конструктора совпадает с именем класса
2. Конструктор не возвращает результат, но и void указывать не надо
3. Конструктор может быть перегружен, причем если появляется какой-то конструктор, который имеет аргументы, то дефолтный конструктор затирается, для его использования нужно написать его явно.

Примеры кода