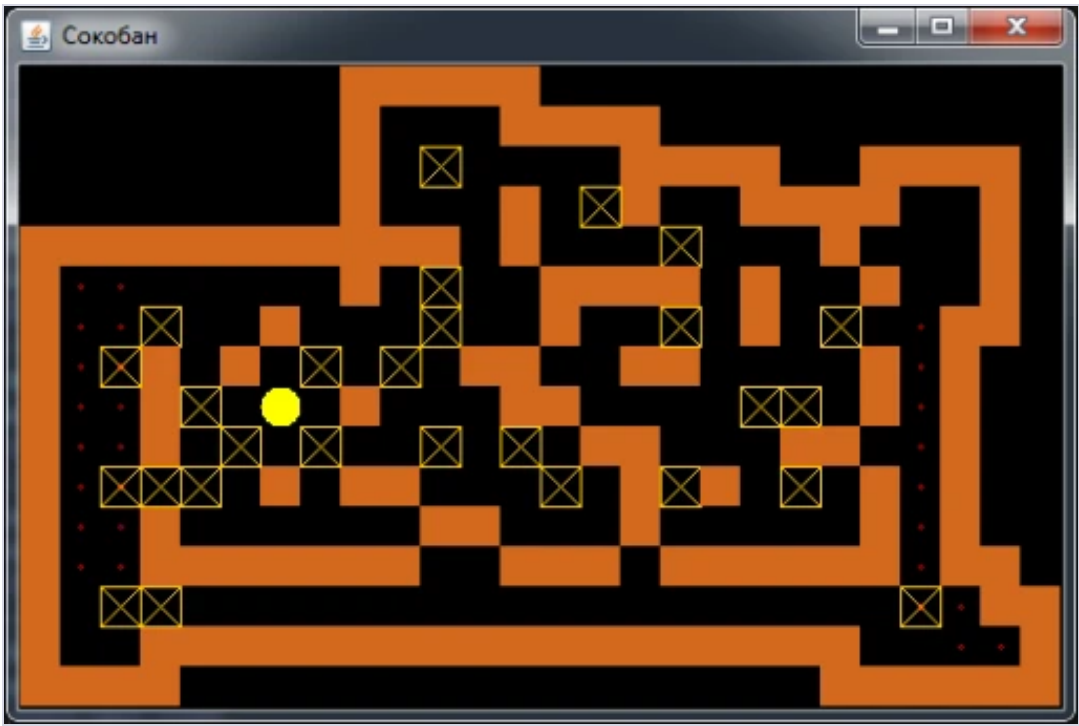


Большая задача: Пишем игру Sokoban

Java Collections
10 уровень, 15 лекция

ОТКРЫТА

- Привет, боец!
- Здравия желаю, Капитан Бобров!
- Поздравляю. Сегодня у нас выходной.
- И мы можем заниматься любыми своими делами?
- Да Амиго, хоть целый день в игрушки шпилить. Например, в Sokoban. Моя самая любимая игра с детства. Вот не могу пройти 435 уровень. Поможешь?
- Конечно помогу, показывайте, товарищ Капитан.
- Капитан Бобров достает из кармана старенький Game boy, запускает Sokoban в монохромном режиме, и тут досада. Батарея разряжена, Game boy выключается.
- Капитан молча смотрит на Game boy, потом на Амиго, потом снова на Game boy, разворачивается и медленно идет в свою комнату.
- Товарищ Капитан, а давайте сами напишем игру Sokoban и придумаем 1000 дополнительных уровней, и крутой графический интерфейс.
- Амиго, ты не перестаешь меня удивлять. Обратись к агенту IntelliJ IDEA. Если у него нет других заданий, и он захочет тебе помочь в свой выходной, тогда приступайте.



Задача  Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА

MEDIUM

Sokoban (1)

★★★★☆☆



Сегодня мы напишем свою реализацию игры Сокобан. Это логическая игра-головоломка, более подробно про нее можешь прочитать в Википедии. Игра будет состоять из 3х основных компонентов (как ты догадался, тут не обошлось без паттерна MVC). Графический интерфейс будет реализован с использованием Swing. 1

Открыть

Задача  Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (2)



Основа заложена, теперь перейдем к ее наполнению. Игровой процесс можно представить как взаимодействие игровых объектов `GameObject`. У нас их будет несколько видов: ящик `Box`, дом `Home` (место куда нужно поместить ящик), стена `Wall` и игрок `Player`.

2.1. Добавь абстрактный класс `GameObject` в пакет `model`.

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (3)

★★★★★

Скоро мы займемся созданием различных игровых объектов. Было бы удобно иметь возможность сразу их где-то нарисовать и посмотреть, как они выглядят. Объекты будут рисоваться на игровом поле `Field`. Тебе пришел измененный код класса `View`, который создает объект поля `Field` и настраивает правильным образом

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (4)

★★★★☆

Общий класс игровых объектов `GameObject` уже есть. Пришло время создать классы конкретных типов игровых объектов.

4.1. Часть игровых объектов могут двигаться (игрок и ящики), а часть нет, например, стены и дома.

4.1.1. Добавь интерфейс `Movable` в пакет `model`. 4.1.2. Интерфейс `Movable` должен иметь мето

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (5)



Игровые объекты “Ящик” и “Игрок” с одной стороны являются объектами, которые могут сталкиваться, а с другой они могут перемещаться по полю. 5.1. Создай класс игрока Player и класс ящика Box в пакете model. Каждый из них унаследуй от максимально подходящего класса. 5.2. Созданные классы должны поддер

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (6)



Пришло время создать класс Home. Этот класс будет отвечать за места на игровом поле (дома) в которые нужно сдвинуть все ящики. Объекты этого типа не должны передвигаться по полю или сталкиваться с другими игровыми объектами.

6.1. Добавь в пакет model класс Home. 6.2. Класс должен быть унаследован от

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (7)

Давай сделаем класс стены Wall. Стена может сталкиваться с другими объектами, но не может двигаться. 7.1. Добавь класс Wall в пакет model. 7.2. Унаследуй класс от подходящего родителя. 7.3. Реализуй конструктор с параметрами int x и int y. 7.4. Реализуй метод отрисовки.

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



EASY



Sokoban (8)



Ты создал полную коллекцию типов игровых объектов. Давай создадим класс, который будет хранить эти объекты.

8.1. Создай класс `GameObjects` в пакете `model`.

8.2. Добавь в него:

- 8.2.1. Поля `Set<Wall> walls`, `Set<Box> boxes`, `Set<Home> homes` и `Player player`.
- 8.2.2. Геттеры для этих полей.
- 8.2.3. Конструктор

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (9)

★★★★★



В процессе работы игры, будут возникать различные события. Давай создадим интерфейс слушателя событий `EventListener`. Его должен реализовывать каждый класс, который хочет обрабатывать события. А классы, которые будут генерировать события, будут вызывать методы этого интерфейса.

9.1. Добавь интерфейс

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (10)



В игре будет несколько уровней, все уровни будут храниться в текстовом файле. Сейчас мы напишем тестовую реализацию загрузчика уровней `LevelLoader`. Почему тестовую? Полный функционал нам пока не нужен, он довольно сложный, оставим его на потом. А пока: 10.1. Создай класс `LevelLoader` в пакете `model`.

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (11)

★★★★★



Начнем наполнять функционалом класс модели Model. Добавь в него: 11.1. Поле GameObjects gameObjects. Оно будет хранить наши игровые объекты. 11.2. Поле отвечающее за текущий уровень int currentLevel. Проинициализируй его значением 1. 11.3. Поле отвечающие за загрузчик уровней LevelLoader levelLoader

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (12)



Попробуем организовать взаимодействие представления и модели. 12.1. Добавь в класс View метод update(), он должен вызывать у игрового поля field метод repaint(). Другими словами, метод update() будет обновлять представление (перерисовывать поле). 12.2. Добавь в класс контроллера метод GameObjects ge

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА



Sokoban (13)



Наполним контроллер функционалом. 13.1. Добавь в конструктор класса `Controller` к тому, что уже есть еще и установку слушателя событий у модели и представления. Слушателем должен быть сам контроллер. 13.2. Реализуй методы контроллера: 13.2.1. `move(Direction direction)` – должен вызывать `move(Direction`

Открыть

Задача Java Collections, 10 уровень, 15 лекция

ЗАКРЫТА





Sokoban (14)



×20

Добавим немного интерактивности в нашу игру (перемещение игрока с помощью клавиатуры). Начнем с обработки нажатия клавиш клавиатуры. 14.1. Добавь в класс Field вложенный класс KeyHandler унаследованный от KeyAdapter. 14.2. Перегрузи у него метод keyPressed(). Если была нажата клавиша с кодом VK_LEFT

Открыть

Sokoban (15)



×40

Пришло время реализовать метод модели, отвечающий за движение `move()`, но для начала реализуем вспомогательные методы. Добавь в класс модели методы: `15.1. boolean checkWallCollision(CollisionObject gameObject, Direction direction)`. Этот метод проверяет столкновение со стеной. Он должен вернуть `true`,

Открыть

Sokoban (16)



×20

Осталось дописать реализацию загрузчика уровней. 16.1. Открой файл levels.txt и внимательно изучи структуру файла. Символ 'X' – означает стену, '*' - ящик, '.' – дом, '&' – ящик который стоит в доме, а '@' - игрока. Всего в файле 60 уровней. 16.2. Реализуй метод GameObjects getLevel(int level). Он д

Открыть

Sokoban (17)



×10

Ты супергерой! Ты сделал отличную игру, можешь немного отдохнуть и поиграть в нее. Если она тебе когда-либо надоест, что практически невозможно, можешь заняться ее улучшениями: 17.1. Использовать картинки для отображения объектов. 17.2. Сделать редактор уровней. 17.3. Добавить рейтинги, можешь даже

Открыть

[← Предыдущая лекция](#)



×31



Комментарии (75)

популярные

НОВЫЕ

старые

JavaCoder

Введите текст комментария

JavaCoder Уровень 50

меньше 20 секунд назад 

Отличная задача. Радует, что практически весь код написан самим, а не как в некоторых больших задачах часть кода дается уже готовым, а то и перетирает уже существующий.

Ответить



Svetlana Vydrina

Уровень 41

15 июля, 23:31

Файл читала с помощью Files.readAllLines()
Удобнее ведь, чем BufferedReader

Ответить

+2

Модератор

Безработный

22 декабря 2021, 19:39

Ну вот поздравляю , ты прошел курс , решай эту задачку и иди во взрослую жизнь

Ответить

+3

Igor Petrashevsky

Уровень 47

31 августа, 02:32

эту задачу решили в технике молодежи в 80х в паре страниц текста

Ответить

0

Модератор

Безработный

31 августа, 14:39

и что?

Ответить

0

Ars

Уровень 41

6 декабря 2021, 18:40

Инфа по последней задаче:
Среднее количество попыток для этой задачи 1.01. Всего эту задачу решили 1501 учеников.

Валидатор хочет, чтобы логика загрузки 1 уровня для 61 была в LoadLevel.getLevel()
Если обнулять уровень в Model, то не пропустит.

Ответить

+2

Модератор

Безработный

22 декабря 2021, 23:22

Среднее количество попыток для этой задачи 1.01. Всего эту задачу решили 1523 учеников.

Ответить

+2

Konstantin

Backend Developer в Google

22 ноября 2021, 16:49

Это я на последней задаче. Сейчас я дома уже.

Ответить

+2

Flexo

Bending Unit #3370318

26 июня 2021, 17:29

Неплохая задача, хотя бы уже тем, что код не подменяется/дописывается сервером.

Немного допилю проект и сделаю сборку, чтобы играть на постоянной)
Нужен хотя бы произвольный выбор уровня

Ответить

+2

tbcarus

Уровень 41, Москва

4 мая 2021, 00:09

В самом конце проявилась допущенная ранее ошибка.

1

Set<Box> setBox = gameObjects.getBoxes();

Почему-то я пропустил, что, создав такой сет, я создал только лишь ещё одну переменную, ссылающуюся на тот же самый сет, и допускал вольности в его отношении.

Ответить

+2

Паровозик, который смог

Уровень 41, Москва, Россия

7 апреля 2021, 23:15

Весьма залипательная игрушка получилась :) Приятно получить такой наглядный и осязаемый результат в итоге :)

Ответить

+2

Aleksandr

Уровень 41

5 апреля 2021, 22:58

Спасибо джавараш за лютую ненависть к 2д играм. Теперь я точно знаю , чем не буду никогда заниматься на джаве.

Ответить

+6

Алексей

Уровень 41, Чебоксары, Россия

21 февраля 2021, 00:03

Интересные задачи, особенно 15 и 16 задачи заставили пошевелить извилинами. Да и с классом Swing поплотнее в этот раз поработали. Хотя что то из графики дали самим сделать. Спасибо проекту!
С валидатором войны не было. Условия тоже максимально понятны.
P.S. Даже открывать следующую лекцию как-то грустно)

Ответить

+3

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

 Русский

▼

