Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**КУРСОВОЙ ПРОЕКТ**

**Дисциплина:** Алгоритмы и структуры данных

**Тема:** Разработка GUI приложения.

Выполнил

студент гр. 3530901/90003 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подольская А.В.

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ахин М.Х.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Санкт-Петербург   
2020

**Оглавление**

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 3](#_Toc58780298)

[МЕТОД РЕШЕНИЯ 4](#_Toc58780299)

[СКРИНШОТЫ ПРОГРАММЫ 6](#_Toc58780300)

[ТЕСТЫ 10](#_Toc58780301)

# **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

Задается какая-то исходная дата (день и месяц). Игра идет с роботом. Каждый игрок на своем ходе задает более позднюю дату, по следующему правилу: увеличивая на 1 или 2 либо день в месяце, либо месяц (но не то и другое сразу). При этом сочетание дня и месяца должно оставаться реальной датой. Игрок, задавший 31.12 побеждает.

В центре находится поле для исходной даты, с которой начинается игра. Ниже находится поле для ввода даты пользователем. Левее находится кнопка «старт» и «сброс». При помощи кнопки «старт» задается исходная дата. При помощи кнопки «сброс» можно начать игру заново. В нижем поле будут записываться ходы игрока и робота. В меню находятся правила игры.

|  |
| --- |
|  |

Github репозиторий проекта: <https://github.com/podolskaya02/Dates>

# **МЕТОД РЕШЕНИЯ**

В проекте использована концепция MVC (Model-View-Controller) для отделения бизнес-логики от визуализации. Код разделен на три пакета классов: model, view, controller.

Пакет model:

1. Содержит класс Fields.java, в котором хранятся три основные поля: для исходной даты, для даты задаваемой пользователем и поле с выводом ходов робота и игрока.

Пакет view:

1. Содержит класс DatesFrame.java, который отвечает за отображения окна со всеми кнопками и полями.
2. Содержит класс MenuForFrame.java, в котором находится меню (правила игры).
3. Содержит класс MessageDialog.java, в котором хранятся три диалоговых окна на случай победы или проигрыша игрока, и нарушения правил.

Пакет controller:

1. Содержит класс Main.java, который отвечает за запуск приложения.
2. Содержит класс MouseListener.java, который обрабатывает события кнопок и полей.
3. Содержит класс CheckDates.java, который рассчитывает возможные ходы игрока. Данный класс проверяет правильность выбора хода пользователем, а также проверяет победил или проиграл игрок.
4. Содержит класс Robot.java, который отвечает за ходы робота.

Два основных алгоритма:

Класс CheckDates метод userMove().

Содержит календарь, куда заносится ход робота. Дату пользователя, которую берут из поля ввода, сравнивают с датой в календаре, к которой прибавляют поочереди день, два дня, месяц, два месяца. Если одна из этих дат эквивалентна дате пользователя, то условие удовлетворено. Запускается метод isRightYear(), который проверяет правильность года. Данная проверка создана для случаев, запрета перехода с ноября и декабря на январь. Если ни одно из условий алгоритма не удовлетворено, пользователю сообщают об ошибки при помощи диалогового окна.

Класс Robot.robotMove().

Содержит календарь, куда заносится ход пользователя. А также же переменную numOfOptions, которое содержит количество вариантов хода робота, она равна четырем. Первым делом алгоритм проверяет, является ли дата пользователя «29.12» или «30.12», для того, чтобы робот мог сделать выигрышный ход. Если эти условия не удовлетворены, то проверяется месяц. В случае, если месяц – ноябрь, то nunOfOptions приравнивается трем, а если декабрь – двум.

numOfOptions:

4 – Прибавить день или два, месяц или два.

3 – Прибавить день или два, или один месяц.

2 - Прибавить день или два.

Таким образом в данной переменной остается то количество вариантов для хода робота, которое не нарушит правило.

Подробная работа программы.

Класс Main.java запускает приложение. Для пользователя открывается окно, на котором он видит поле для исходной даты, кнопку «старт», заблокированное поле для дальнейших дат игрока, заблокированную кнопку «сброс», поле для вывода ходов игрока и робота, и меню. В меню игрок может ознакомиться с правилами игры.

Класс MenuForFrame содержит метод setMenu(), куда заносятся правила и подключается слушатель. Данный метод выводит правила игры на экран, после нажатия на «правила».

В поле для исходной даты игрок при помощи календаря выбирает любую дату, с которой он хочет начать игру, а после нажимает кнопку «старт». MouseListener.addStartFieldListener() обрабатывает данные события: выводит значение исходной даты, блокирует кнопку старт, включает кнопку «сброс», разрешает пользование поля для дальнейших дат игрока, вычеркивает из календаря все даты до выбранной пользователем. Передает ход роботу.

В классе Robot.java метод moveForStartField() считывает дату из стартового поля, и выводит ход робота, который рассчитывается при помощи метода robotMove(), а метод CheckDates.isUserLose() проверяет проиграл ли игрок после этого хода робота.

Далее ход пользователя. Игрок открывает календарь, и выбирает любую дату соответствующую правилам. MouseListener.addDataFieldListener() считывает данные и передает в класс CheckDates, где метод userMove() проверяет правильность хода игрока. Если дата неверна – метод сообщает об этом в класс MessageDialog, после чего метод wrongDatesMessage() выводит на экран соответствующее сообщение. Если дата верна, то запускается CheckDates.isUserWin(), где данный метод проверяет, победил ли пользователем своим ходом. Если дата верна и пользователь еще не победил, то ход передает роботу.

Robot.moveForDataField() считывает ход игрока, запускает метод robotMove() и метод isUserLose(), выводит ход робота.

Победа пользователя проверяется в классе CheckDates при помощи метода isUserWin(). Если дата соответствует 31.12 – класс MessageDialog с помощью метода winMessage() выводит соответствующее сообщение. А поле для ввода даты блокируется.

Проигрыш пользователя проверяет в классе CheckDates при помощи метода isUserLose(). Если после хода робота дата соответствует 31.12 – класс MessageDialog с помощью метода loseMessage() выводит соответствующее сообщение. А поле для ввода даты блокируется.

В классе CheckDates проверяется, чтобы дата отличалась от предыдущей на 1 или 2 дня, на 1 или 2 месяца (один вариант из 4), также проверяется, чтобы игрок не перешел на следующий год.

Кнопка «сброс» - позволяет начать игру заново в любой момент. MouseListener.addStopButtonListener() очищает поле с выводами ходов, блокирует поле для даты игрока, разрешает пользование кнопкой «старт» и полем для исходной даты.

Для прикрепления календаря к полям исходной даты и даты пользователя использовался JXDatePicker. Для считывания и расчета дат использовались классы Date и Calendar. Для отображения даты в удобном формате использовался SimpleDateFormat.

# **СКРИНШОТЫ ПРОГРАММЫ**

|  |
| --- |
|  |

Рис.1 Исходный вид окна

|  |
| --- |
|  |

Рис. 2 Изменения внешнего вида окна после выбора исходной даты

|  |
| --- |
|  |

Рис 3. Ход игрока

|  |
| --- |
|  |

Рис. 4. Вывод wrongDateMessage

|  |
| --- |
|  |

Рис. 5. Вывод winMessage

|  |
| --- |
|  |

Рис. 6. Вывод loseMessage

|  |
| --- |
|  |

Рис. 7. Правила игры

# **ТЕСТЫ**

Для проверки корректной работы программы были созданы тесты.

1. UserMoveTest, проверяющий правильность хода игрока. На вход был подан ход робота в виде даты. Тест включает проверку всех 4 возможных дат пользователя, которые должен рассчитать метод userMove(). Проверка идет при помощи метода assertTrue(), тест включает проверку неправильных дат при помощи метода assertFalse(). Также в данном тесте проверяется метод isRightYear(), который предназначен для запрета перехода на следующий год.
2. IsUserWinTest проверяет, победил ли игрок. На вход подается ход пользователя. Дата должна соответствовать 31.12
3. IsUserLoseTest проверяет, проиграл ли игрока. На вход подает ход робота. Дата должна соответствовать 31.12
4. RobotMoveTest проверяет правильность расчетов хода робота. На вход подается дата пользователя. Тест проверяет все возможные варианты хода робота, который должен рассчитать метод robotMove(), учитывая все ограничения.