**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Профессор  департамента программной инженерии факультета компьютерных наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. М. Баканов «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2019 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Шилов «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

|  |  |
| --- | --- |
| ***Подп. и дата*** |  |
| ***Инв. № дубл.*** |  |
| ***Взам. инв. №*** |  |
| ***Подп. и дата*** |  |
| ***Инв. № подл*** | RU.17701729.04.01-01 81 01-1-ЛУ |

**ПРОГРАММА МОДЕЛИРОВАНИЯ И ВЫБОРА ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ДВИЖЕНИЯ ПЛОСКИХ МЕХАНИЗМОВ**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.01-01 81 01-1\_ЛУ**

Исполнитель  
студент группы БПИ172  
\_\_\_\_\_\_\_\_\_\_ / Пучков А. М. /  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2019 г.

**Москва 2019**

УТВЕРЖДЕНRU.17701729.04.01-01 81 01-1-ЛУ

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл | RU.17701729.04.01-01 81 01-1 |

**ПРОГРАММА МОДЕЛИРОВАНИЯ И ВЫБОРА ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ДВИЖЕНИЯ ПЛОСКИХ МЕХАНИЗМОВ**

**Пояснительная записка**

**RU.17701729.04.01-01 81 01-1**

**Листов 33**

**Москва 2019**

Содержание

[1. Введение 3](#_Toc482222293)

[1.1. Наименование программы 3](#_Toc482222294)

[1.2. Основания для разработки 3](#_Toc482222295)

[2. Назначение и область применения 4](#_Toc482222296)

[2.1. Назначение разработки 4](#_Toc482222297)

[2.1.1. Функциональное назначение 4](#_Toc482222298)

[2.1.2. Эксплуатационное назначение 4](#_Toc482222299)

[2.2. Область применения 4](#_Toc482222300)

[3. Технические характеристики 5](#_Toc482222301)

[3.1. Постановка задачи на разработку программы 5](#_Toc482222302)

[3.2. Описание и обоснование алгоритма и функционирования программы 5](#_Toc482222303)

[3.2.1. Описание алгоритма и функционирования программы 5](#_Toc482222304)

[3.2.2. Обоснование выбора алгоритма 6](#_Toc482222305)

[3.3. Описание и обоснование метода организации входных и выходных данных 7](#_Toc482222306)

[3.3.1. Описание метода организации входных и выходных данных 7](#_Toc482222307)

[3.3.2. Обоснование метода организации входных и выходных данных 7](#_Toc482222308)

[3.4. Описание и обоснование выбора состава технических и программных средств 7](#_Toc482222309)

[3.4.1. Состав технических и программных средств 7](#_Toc482222310)

[3.4.2. Обоснование выбора состава технических и программных средств 8](#_Toc482222311)

[4. Технико-экономические показатели 9](#_Toc482222312)

[4.1. Предполагаемая потребность 9](#_Toc482222313)

[4.2. Ориентировочная экономическая эффективность 9](#_Toc482222314)

[4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 9](#_Toc482222315)

[Приложение 1. Список используемой литературы 10](#_Toc482222316)

[Приложение 2. Описание и функциональное назначение классов 11](#_Toc482222317)

[Приложение 3. Описание и функциональное назначение полей и методов 13](#_Toc482222318)

# 1. Введение

## 1.1. Наименование программы

Наименование программы: «Программа моделирования и выбора оптимальных параметров движения плоских механизмов» («The simulation program and the selection of optimal parameters of the plane mechanisms»).

## 1.2. Основания для разработки

Основанием для разработки программы является Приказ декана ФКН И. В. Аржанцева № 2.3-02/1012-0 2 от 10.12.2018 "Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук".

Программа разрабатывается в рамках выполнения курсовой работы по теме «Программа моделирования и выбора оптимальных параметров движения плоских механизмов».

# 2. Назначение и область применения

## 2.1. Назначение разработки

### 2.1.1. Функциональное назначение

Функциональным назначением программы является моделирование и оптимизация плоского механизма, и вывод модели на экран. В течение сеанса работы с программой пользователь может:

1. Видеть текущее положение механизма
2. Управлять скоростью и направлением движения механизма
3. Выбирать конфигурацию демонстрируемого программой механизма из предложенных.

Программа представляет собой десктопное Windows-приложение.

### 2.1.2. Эксплуатационное назначение

Эксплуатационным назначением данной программы является её использование при исследовании и оптимизации плоских механизмов с использованием численных методов.

## 2.2. Область применения

Предназначена для численного моделирования механики и построения модели плоского механизма, оптимизации по нескольким параметрам, и графической интерпретации результатов расчета.

# 3. Технические характеристики

## 3.1. Постановка задачи на разработку программы

Программа должна обеспечивать возможность выполнения следующих функций:

1. Построение механизма предопределенной конфигурации и отрисовка его на экране;
2. Визуализация движения механизма, изменение скорости и направления движения;
3. Создание списка механизмов с возможностью выбора пользователем его длины;
4. Чтение механизма из списка по номеру и вывод его на экран;
5. Оптимизация параметров механизма через линейную регрессию и градиентный спуск.

## 3.2. Описание и обоснование алгоритма и функционирования программы

### 3.2.1. Описание алгоритма и функционирования программы

Движение механизма осуществляется следующим образом: правое короткое звено (далее приводящий кривошип) поворачивается относительно своего начала (красная точка) на угол, выбранный по положению соответствующего бегунка. Затем из возможных положений левого короткого звена, полученных виртуальным поворачиванием последнего и сохранением координат его конца, выбирается такое, чтобы разница расстояния от него до конца приводящего кривошипа и половины длины длинного кривошипа (единственного со свободным концом) была минимальна, затем свободный конец длинного кривошипа приводится на одну линию с его началом и серединой на расстоянии половины длины от середины. Координаты конца записываются в список позиций, и затем используются для визуализации траектории.

Для создания списка механизмов используется генератор псевдорандомных чисел, выдающий 4 значения: длины 3 кривошипов, и расстояние между фиксированными точками механизма – началами приводящего и короткого левого звена. Затем значения проверяются на возможность существования механизма с такими параметрами, и, если проходят проверку, механизм записывается в файл, а список его положений (координаты свободного конца) при вращении записываются в другой файл.

Для нахождения оптимальных параметров используется линейная регрессия и градиентный спуск. Сначала, для каждого файла с набором точек считается аппроксимирующая прямая вида y = k\*x + b (с помощью линейной регрессии). Затем сохраняется среднеквадратичная ошибка для этой прямой, которая сопоставляется с набором параметров механизма, создавшего эти точки. На таких наборах проводится новая линейная регрессия, на выходе которой имеется функция вида , по которой проводится градиентный спуск с ограничениями на параметры. По результатам последнего действия строится механизм, считающийся оптимальным.

### 3.2.2. Обоснование выбора алгоритма

Несмотря на то, что задача оптимизации плоских механизмов (и лямбда-механизма, в частности) была аналитически решена еще П. Чебышёвым, его решение не является масштабируемым, в отличие от численных методов, применяемых в программе. Таким образом, при возрастании сложности механизма, единственным известным сейчас методом оптимизации будет являться численный метод, примененный в программе.

## 3.3. Описание и обоснование метода организации входных и выходных данных

### 3.3.1. Описание метода организации входных и выходных данных

#### 3.3.1.1. Описание метода организации входных данных

Программа позволяет пользователю выбирать параметры движения механизма (его наличие, скорость, направление), наличие отрисовки траектории свободного конца механизма, выполнение оптимизации механизма. Данные функции реализованы с помощью стандартных элементов Java SwingFrame: кнопок JButton, ползунков JSlider и текстового поля JTextField.

#### 3.3.1.2. Описание метода организации выходных данных

Программа выводит положение механизма в поле JPanel, а текстовые данные – в стандартную консоль ОС Windows.

### 3.3.2. Обоснование метода организации входных и выходных данных

#### 3.3.2.1. Обоснование метода организации входных данных

Примененный способ ввода данных и управления признан наиболее удобным, так как отделяет управление движением механизма в верхней части окна приложения от управления его оптимизацией в нижней.

#### 3.3.2.2. Обоснование метода организации выходных данных

Данный метод вывода (разделение текста и изображений в разные окна) избран как наиболее простой для восприятия и не загромождающий интерфейс приложения способ передачи информации пользователю через различные каналы.

## 3.4. Описание и обоснование выбора состава технических и программных средств

### 3.4.1. Состав технических и программных средств

#### 3.4.1.1. Состав технических средств

В состав технических средств должен входить персональный компьютер со следующими характеристиками:

1. 4 Гб и более оперативной памяти
2. монитор, разрешение экрана 3200х1800 пикселей.
3. 32-разрядный или 64-разрядный процессор intel core i7 с тактовой частотой 2.9 ГГц и выше;
4. клавиатура, мышь или трекпад.

#### 3.4.1.2. Состав программных средств

Для работы программы необходим следующий состав программных средств:

1) операционная система Microsoft Windows 10 или более поздняя версия;

2) Java Virtual Machine 8.0

3) Интерпретатор языка Python версии 3.x или новее

4) Библиотеки Python pandas, numpy, sklearn

### 3.4.2. Обоснование выбора состава технических и программных средств

#### 3.4.2.1. Обоснование выбора состава технических средств

При процессоре медленнее указанного скорость работы приложения может значительно падать.

Разрешение экрана, отличное от указанного, вызовет неправильное расположение элементов интерфейса и может повлиять на восприятие программы пользователем.

Программа в большинстве случаев использует значительный объем памяти для чтения из файла, записи в него или проведения расчетов с матрицами. При недостаточном свободном объеме ОЗУ работа приложения может нарушиться, а используемые данные – быть поврежденными.

Программа использует функции, реализованные в соответствующих библиотеках языка Python, при их отсутствии некоторые функции будут недоступны.

При размере и разрешении дисплея менее указанного интерфейс программы может некорректно отображаться.

#### 3.4.2.2. Обоснование выбора состава программных средств

ОС Windows 7 была выбрана, как наиболее популярная в недавнем прошлом (43% на июнь 2018-го), а последующие версии – 8 и 10 в сумме дают еще 35% охвата ПК (в июне 2018).

# 4. Технико-экономические показатели

## 4.1. Предполагаемая потребность

Данная программа ориентируется на людей, заинтересованных в работе или исследованиях, связанных с плоскими механизмами. Программа может быть использована в учебных целях для демонстрации работы алгоритмов численной оптимизации и движения плоских механизмов.

## 4.2. Ориентировочная экономическая эффективность

В рамках данной работы расчет экономической эффективности не предусмотрен.

## 4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

Быстрый поиск в сети Интернет полных аналогов данной программы не обнаружил.

Преимуществами данной программы являются:

- открытый код,

- свободное распространение,

- понятный интерфейс.

# Приложение 1. Список используемой литературы

1. Хорстманн К., Корнелл Г. Java. Библиотека профессионала, том 1. Основы. 9-е изд.: пер. с англ. – М.: ООО «И. Д. Вильямс», 2014.
2. Java™ Platform, Standard Edition 8 API Specification // Oracle – [Электронный ресурс]. URL: <https://docs.oracle.com/javase/8/docs/api/index.html> (дата обращения: 4.12.2018 г.)
3. Android API Reference // Android Developers – [Электронный ресурс]. URL: <https://developer.android.com/reference/> (дата обращения: 25.03.2019 г.)
4. Единая система программной документации: сборник, офиц. изд. – М.: ИПК Издательство стандартов, 2001.
5. Новожилов И., Зацепин М. Типовые расчеты по теоретической механике на базе ЭВМ – М: «Высшая школа», 1986.
6. Станок ХПЛ-650 – [Электронный ресурс]. URL: <http://vbakanov.ru/right_5.htm>, дата обращения (10.05.2019 г.)

# Приложение 2. Описание и функциональное назначение классов

Описываются только классы, содержащие вручную написанные поля и методы. Полностью автоматически сгенерированные классы не описываются.

*Табл. 1. Описание и функциональное назначение классов*

|  |  |
| --- | --- |
| Класс | Назначение |
| Graphics.DrawPanel | Представляет собой поле, на котором рисуется механизм |
| Graphics.MechControls | Верхнее меню управления движением механизма |
| Graphics.OptimisationControls | Нижнее меню управления оптимизацией |
| Graphics.Swing\_Frame | Окно приложения |
| Helpers.DoublePair | Класс-пара чисел с плавающей точкой |
| Helpers.Utils | Класс со статическими методами, упрощающими работу в приложении |
| Helpers.WriterReader | Класс с функциями записи в и чтения из файла |
| Mechanics.Edge | Звено механизма |
| Mechanics.Mechanism | Реализация самого механизма |
| Mechanics.Mechpart | Интерфейс, реализуемый звеньями приложения |
| Mechanics.MyPoint | Сочленение механизма |
| Mechanics.TriEdge | Звено механизма с тремя точками крепления |
| Optimisation.LegionCreator | Класс, создающий список автоматически генерируемых механизмов |
| Optimisation.OutsourceCreator | Выполняет запуск .ру-файла с математическими функциями |

# Приложение 3. Описание и функциональное назначение полей и методов

Описываются только вручную написанные поля и методы. Полностью автоматически сгенерированные поля и методы не описываются.

*Табл. 2. Описание и функциональное назначение полей класса DrawPanel*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| drawPath | Package-private | boolean | Отвечает за необходимость прорисовки пути свободного конца механизма |
| mech | Package-private | Mechanism | Рисуемый механизм |
| positions | Package-private | LinkedList<Point2D> | Координаты положений свободного конца механизма |

*Табл. 3. Описание и функциональное назначение методов класса DrawPanel*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| DrawPanel | Package-privvate | - | Mechanism | Конструктор, присваивает полученный механизм в соответствующее поле |
| updateMech | Package-privvate | void | Mechanism | Присваивает в поле mech полученный механизм |
| paint | public | void | Graphics | Перерисовывает механизм и, при необходимости, позиции его свободного конца |
| drawMech | private | void | Graphics2D, Mechanism | Прорисовывает сначала кривошипы, а затем сочленения механизма |
| drawPath | private | void | Graphics2D | Прорисовывает положения свободного конца механизма, сохраненные в positions |

*Табл. 4. Описание и функциональное назначение полей класса MechControls*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| mech | private | Mechanism | Текущий управляемый механизм |
| drawPanel | private | DrawPanel | Объект DrawPanel, на который происходит воздействие |
| rotateOn | private | boolean | Поле, показывающее необходимость самостоятельного движения механизма |
| sliderLabel | private | JLabel | Левая надпись около бегунка, показывает минимальное значение угла вращения |
| Slider | private | JSlider | Бегунок, отвечает за угол вращения |
| rightLabel | private | JLabel | Надпись справа от бегунка, показывает максимальный угол вращения |
| rotBut | private | JButton | Кнопка «повернуть единожды» |
| showPath | private | JButton | Кнопка, включающая отрисовку пути свободного конца механизма |
| deletePath | private | JButton | Кнопка, отключающая отрисовку пути свободного конца механизма |
| startRotating | private | JButton | Кнопка, включающая автоматическое вращение механизма |
| stopRotating | private | JButton | Кнопка, останавливающая автоматическое вращение |

*Табл. 5. Описание и функциональное назначение методов класса MechControls*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| MechControls | Package-private | - | Mechanism, DrawPanel | Конструктор, инициализирует все поля и присваивает элементам контроля соответствующие действия |
| addPoint | Package-private | void | int, LinkedList<Point2D> | Добавляет точку с указанным номером в переданный список или перезаписывает существующую |

*Табл. 6. Описание и функциональное назначение полей класса OptimisationControls*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| drawPanel | private | DrawPanel | Объект DrawPanel, на который происходит воздействие |
| sliderLabel | private | JLabel | Надпись слева от ползунка, показывает минимальный размер списка |
| slider | private | JSlider | Ползунок, отвечает за размер создаваемого списка механизмов |
| rightLabel | private | JLabel | Надпись справа от ползунка, показывает максимальный размер списка |
| createTrainSamples | private | JButton | Кнопка, инициализирующая создание списка |
| num | private | JTextField | Текстовое поле для ввода номера загружаемого механизма |
| loadMech | Pprivate | JButton | Кнопка, загружающая механизм под введенным номером |
| train | private | JButton | Кнопка, запускающая регрессию |

*Табл. 7. Описание и функциональное назначение методов класса OptimisationControls*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OptimisationControls | Package-private | - | DrawPanel | Конструктор |
| loadMech | private | void | - | Проверяет корректнсть введенных в текстовое поле данных, и выполняет загрузку механизма и обновление его в drawPanel |

*Табл. 8. Описание и функциональное назначение полей класса SwingFrame*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| canvas | private | DrawPanel | Поле для рисования |
| gui | private | MechControls | Панель управления движением механизма |
| Opt | private | OptimisationControls | Панель управления оптимизацией механизма |
| mech | private | Mechanism | Управляемый механизм |

*Табл. 9. Описание и функциональное назначение методов класса SwingFrame*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| createMech | private | Mechanism | - | Создает базовый механизм |
| main | public | void | String… | Основная точка входа в программу |
| SwingFrame | private | - | - | Конструктор |

*Табл. 10. Описание и функциональное назначение полей класса DoublePair*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| first | public | double | Первый элемент пары |
| second | public | double | Второй элемент пары |

*Табл. 11. Описание и функциональное назначение методов класса DoublePair*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| DoublePair | public | - | double… | Конструктор |
| toString | public | String | - | Возвращает строковое представление пары |
| add | public | DoublePair | DoublePair | Суммирует значения двух пар |

*Табл. 12. Описание и функциональное назначение методов класса Utils*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| distance | public | double | DoublePair… | Возвращает расстояние между точками, заданными парами. |
| distance | public | double | DoublePair, MyPoint | Возвращает расстояние между точками, заданными парой и точкой. |
| distance | public | double | MyPoint… | Возвращает расстояние между точками. |
| Pif2Pol | public | DoublePair | DoublePair | Переводит координаты точки из ортогональной системы в полярную |
| Pol2Pif | public | DoublePair | DoublePair | Переводит координаты точки из полярной системы в ортогональную |
| asDegrees | public | double | double | Переводит угол из радиан в градусы |
| asRadians | public | double | double | Переводит угол из градусов в радианы. |
| distanceToLine | public | double | double… | Возвращает расстояние от точки до прямой |
| distanceEquals | public | boolean | MyPoint, DoublePair, double | Показывает, совпадает ли расстояние от точки до прямой с переданным значением с некоторой точностю |

*Табл. 13. Описание и функциональное назначение полей класса WriterReader*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| path | private | String | Путь к файлу для чтения/записи |
| features | public | double[][] | Прочитанная таблица |

*Табл. 14. Описание и функциональное назначение методов класса WriterReader*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| WriteReader | public | - | String | Конструктор. |
| readSingle | public | void | - | Читает данные из файла в таблицу |
| write | public | void | double…, double | Записывает данные в файл |
| write | public | void | double… | Записывает таблицу в файл. |

*Табл. 15. Описание и функциональное назначение полей класса Edge*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| beginning | private | MyPoint | Точка-начало кривошипа |
| end | private | MyPoint | Точка-конец кривошипа |
| possiblePositions | private | DoublePair | Массив возможных положений конца кривошипа |
| length | private | double | Длина кривошипа |

*Табл. 16. Описание и функциональное назначение методов класса Edge*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Length | public | double | - | Свойство длины кривошипа. |
| Angle | public | double | - | Свойство, возвращает угол наклона относительно горизонтали |
| setLength | public | void | double | Устанавливает значение длины кривошипа |
| toString | public | String | - | Возвращает строковое представление кривошипа |
| getBeginning | public | MyPoint | - | Возвращает точку начала |
| getEnd | public | MyPoint | - | Возвращает точку конца |
| getStartX | public | double | - | Возвращает абсциссу начала |
| setStartX | private | void | double | Устанавливает абсциссу начала |
| getStartY | public | double | - | Возвращает ординату начала |
| setStartY | private | void | double | Устанавливает ординату начала |
| getEndX | public | double | - | Возвращает абсциссу конца |
| setEndX | private | void | double | Устанавливает абсциссу конца |
| getEndY | public | double | - | Возвращает ординату конца |
| setEndY | private | void | double | Устанавливает ординату конца |
| setStart | public | void | double… | Устанавливает координаты начала |
| setFinish | public | void | double… | Устанавливает координаты конца |
| getStart | public | DoublePair | - | Возвращает пару-координаты начала |
| getFinish | public | DoublePair | - | Возвращает пару-координаты конца |
| getCenter | public | DoublePair | - | Заглушка для наследования интерфейса |
| Edge | Package-private | - | MyPoint… | Конструктор |
| recalculatePositions | public | void | - | Расчитывает все возможные положения конца кривошипа |
| move | public | void | double… | Перемещает кривошип по переданным изменениям каждой из координат |
| move | public | void | double… | Перемещает кривошип по изменениям абсцисс и ординат |
| rotate | public | void | double | Поворачивает кривошип вокруг точки начала на заданный угол |

*Табл. 17. Описание и функциональное назначение полей класса Mechanism*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| contents | public | MechPart… | Массив кривошипов |
| joints | public | MyPoint… | Массив сочленений |
| positions | public | LinkedList<Point2D> | Список положений свободного конца механизма |

*Табл. 18. Описание и функциональное назначение методов класса Mechanism*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Mechanism | public | - | LinkedList<int[]>, MyPoint… | Конструктор |
| createMech | public | Mechamism | double… | Создает механизм по заданным параметрам |
| moveMech | public | void | double | Поворачивает приводящий кривошип на заданное значение в градусах |
| toArray | public | double… | - | Представляет механизм в виде набора длин кривошипов |
| checkFunctionality | public | boolean | - | Показывает, способен ли механизм совершить полный оборот без нарушения длин своих кривошипов. |
| initPositions | public | void | - | Проворачивает приводящий кривошип на 360 градусов и записывает положения свободного конца. |

*Табл. 19. Описание и функциональное назначение полей класса MyPoint*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| constant | private | boolean | Показывает, является ли точка неподвижной |
| centerX | private | double | Абсцисса центра |
| cenretY | private | double | Ордината центра |

*Табл. 20. Описание и функциональное назначение методов класса MyPoint*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| getCenterX | public | double | - | Возвращает абсциссу центра. |
| setCenterX | Package-private | void | double | Обновляет абсциссу центра переданным значением. |
| getCenterY | public | double | - | Возвращает ординату центра |
| setCenterY | Package-private | void | double | Обновляет ординату центра переданным значением |
| MyPoint | Package-private | - | - | Конструктор по умолчанию. |
| MyPoint | public | - | double… | Конструктор по 2 координатам |
| MyPoint | public | - | double…, boolean | Конструктор по координатам и показателю движимости точки |
| move | Package-private | void | double… | Сдвигает точку по переданным изменениям координат |
| isConstant | public | boolean | - | Возвращает невозможность точки к движению |

*Табл. 21. Описание и функциональное назначение полей класса TriEdge*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| first | private | Edge | Часть кривошипа от начала к середине |
| second | private | Edge | Часть кривошипа от середины к концу |
| main | private | Edge | Основная часть кривошипа, от начала к концу |
| beginning | private | MyPoint | Точка-начало |
| finish | private | MyPoint | Точка-конец |
| center | private | MyPoint | Точка-середина |
| possibleFinishPositions | private | DoublePair… | Массив положений конца |
| possibleMiddlePositions | private | DoublePair… | Массив положений середины |

*Табл. 22. Описание и функциональное назначение методов класса TriEdge*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| TriEdge | Packadge-private | - | MyPoint… | Конструктор. |
| getStartX | public | Round | - | Абсцисса начала |
| getStartY | public | boolean | - | Ордината начала |
| getEndX | public | boolean | - | Абсцисса конца |
| getEndY | public | boolean | - | Ордината конца |
| getBeginning | public | MyPoint | - | Точка начала |
| getMiddle | private | MyPoint | - | Точка середины |
| getEnd | public | MyPoint | - | Точка конца |
| setCenter | private | void | double… | Устанавливает середину |
| getCenterX | public | double | - | Устанавливает абсциссу середины |
| getCenterY | public | double | - | Устанавливает ординату середины |
| getCenter | public | DoublePair | - | Возвращает пару-центр |
| setStart | public | void | double… | Устанавливает начало |
| setFinish | public | void | double… | Устанавливает конец |
| setFinish | public | void | int | Устанавливает конец по индексу в массиве possibleFinishPositions |
| getStart | public | DoublePair | - | Возвращает пару-начало |
| getFinish | public | DoublePair | - | Возвращает пару-конец |
| move | public | void | double… | Двигает по отдельным изменениям координат |
| rotate | public | void | double | Поворацивает вокруг начала на значение градусов |
| move | public | void | double… | Двигает на изменение абсциссы и ординаты |
| inline | public | void | - | Устанавливает центр ровно посередине между началом и концом |
| inlineFinish | package-private | void | - | Устанавливает конец на одной линии с началом и серединой, на расстоянии половины длины от середины |
| Angle | public | double | - | Возвращает угол с горизонталью |
| Length | public | double | - | Длина от начала до конца |
| setLength | public | void | double | Устанавливает длину |
| recalculatePositions | public | void | - | Пересчитывает массивы possibleFinishPositions и possibleMiddlePositions |
| getPointWithDistance | public | int | MechPart, boolean, double | Возвращает индекс позиции, наиболее точно отстоящей от конца MechPart на дистанцию. Если 2-й аргумент true, позиция конца, иначе - середины |

*Табл. 23. Описание и функциональное назначение полей класса LegionCreator*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Назначение |
| ran | private | Random | Генератор псевдослучайных чисел |

*Табл. 24. Описание и функциональное назначение методов класса LegionCreator*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| LegionCreator | public | - | String, int | Конструктор, запускает создание списка механизмов |

*Табл. 25. Описание и функциональное назначение методов класса OutsourceAttarctor*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Outsource Attarctor | public | - | - | Конструктор по умолчанию. |
| runPython | public | void | - | Запускает .py файл с регрессией |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Изм. | Номера листов | | | | Всего листов (страниц) в документе | № документа | Входя- щий № сопро- водит. докум. и дата | Под- пись | Дата |
| изменен- ных | заменен- ных | новых | аннулиро- ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |