**1ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ   
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Преподаватель департамента программной инженерии факультета компьютерных наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.Н. Береснева  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. | УТВЕРЖДАЮ  Академический руководитель  образовательной программы  «Программная инженерия», канд. техн.  наук, профессор ДПИ ФКН  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г. |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. Инв. № |  | | Подп. и дата |  | | Инв. № подл. |  | | **ПРОГРАММА РЕШЕНИЯ ЗАДАЧИ МАРШРУТИЗАЦИИ С ОГРАНИЧЕНИЕМ ПО ГРУЗОПОДЪЕМНОСТИ НА ОСНОВЕ МЕТОДА ИМИТАЦИИ ОТЖИГА**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ RU.17701729.503200-01 81 01-1-ЛУ**  Исполнитель  Студент группы БПИ163  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Д.В. Строков /  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018 г. | |  |

**УТВЕРЖДЕН**

**RU.17701729.503200-01 81 01-1 ЛУ**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. Инв. № |  | | Подп. и дата |  | | Инв. № подл. |  | | **ПРОГРАММА РЕШЕНИЯ ЗАДАЧИ МАРШРУТИЗАЦИИ С ОГРАНИЧЕНИЕМ ПО ГРУЗОПОДЪЕМНОСТИ НА ОСНОВЕ МЕТОДА ИМИТАЦИИ ОТЖИГА**  **Пояснительная записка  RU.17701729.503200-01 81 01-1  Листов 38** |  |

**СОДЕРЖАНИЕ**

[1. ВВЕДЕНИЕ 3](#_Toc483276326)

[1.1. Наименование программы 3](#_Toc483276327)

[1.2. Документы, на основании которых ведётся разработка 3](#_Toc483276328)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 4](#_Toc483276329)

[2.1. Функциональное назначение 4](#_Toc483276330)

[2.2. Эксплуатационное назначение 4](#_Toc483276331)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 5](#_Toc483276332)

[3.1. Постановка задачи на разработку программы 5](#_Toc483276333)

[3.2. Описание алгоритмов и функционирования программы 5](#_Toc483276334)

[3.2.1 Переходы между сценами 5](#_Toc483276335)

[3.2.2 Генерация игрового поля 5](#_Toc483276336)

[3.2.3 Генерация графа исходов для искусственного интеллекта 6](#_Toc483276337)

[3.2.4 Ход искусственного интеллекта 7](#_Toc483276338)

[3.3 Описание и обоснование выбора метода организации входных и выходных данных 7](#_Toc483276339)

[3.4 Описание и обоснование выбора состава технических и программных средств 7](#_Toc483276340)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 8](#_Toc483276341)

[4.1. Ориентировочная экономическая эффективность 8](#_Toc483276342)

[4.2. Предполагаемая потребность 8](#_Toc483276343)

[4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 8](#_Toc483276344)

[5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 9](#_Toc483276345)

[Приложение 1 10](#_Toc483276346)

[Приложение 2 12](#_Toc483276347)

[Приложение 3 14](#_Toc483276348)

# 1. ВВЕДЕНИЕ

## 1.1. Наименование программы

Наименование программы: «Программа решения задачи маршрутизации с ограничением по грузоподъемности на основе метода имитации отжига» («Program for Solving the Capacitated Vehicle Routing Problem Based on Simulated Annealing Algorithm»).

## 1.2. Документы, на основании которых ведётся разработка

Приказ НИУ ВШЭ об утверждении тем курсовых работ № 2.3-02/1502-01 от 15.02.2017, утверждённый руководителем департамента программной инженерии факультета компьютерных наук.

# 2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

## 2.1. Функциональное назначение

Функциональным назначением программы является получение входных данных для задачи маршрутизации с ограничением по грузоподъёмности и решение данной задачи с использованием метода имитации отжига, а также наглядная визуализация работы данного алгоритма и вывод результатов его работы.

## 2.2. Эксплуатационное назначение

Эксплуатационным назначением программы является её использование для решения задач маршрутизации с ограничением по грузоподъёмности, и получения наглядного отображения работы алгоритма имитации отжига. Также программа может пригодиться для практических целей в логистике – для подбора оптимального маршрута доставки грузов.

# 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

## 3.1. Постановка задачи на разработку программы

Программа должна посредством алгоритма имитации отжига на заданном наборе вершин с двумерными координатами и заданным в виде числа грузом для каждой вершины, а также числе максимальной нагрузки на транспорт и номеру вершины, являющейся депо решать и визуализировать процесс решения задачи маршрутизации с ограничением грузоподъёмности – строить набор маршрутов, минимизируя их суммарную длину, так чтобы каждая вершина (кроме вершины - депо) была посещена ровно один раз, суммарная нагрузка на каждом маршруте не превышала заданную максимальную нагрузку, а каждый маршрут начинался и заканчивался в вершине - депо. Также должна быть возможно сохранять и загружать готовые решения.

## 3.2. Описание алгоритмов и функционирования программы

### 3.2.1 Общая структура программы

Модуль исполнения служит точкой запуска программы. После запуска интерфейса модуль исполнения принимает команды интерфейса на запуск, паузу или остановку алгоритма имитации отжига, а также на сохранение и загрузку файлов и отправляет текущее состояние выполнения задачи. Модуль интерфейса получает от пользователя вышеперечисленные команды и отправляет их в модуль исполнения, а также визуализирует текущее состояние.

### 3.2.2 Алгоритм имитации отжига

Изначально алгоритм генерирует случайное состояние – набор маршрутов который не нарушает наложенные ограничения, а также устанавливает заданную начальную температуру. Затем происходит последовательное исполнение итераций.

На каждой итерации генерируется “соседнее” состояние следующим образом: случайным образом выбираются от 1 до 8 вершин и перемещаются в случайное место таким образом, чтобы не нарушались наложенные ограничения. Пусть delta = разность суммарных расстояний маршрутов в новом и текущем состояниях. Тогда вероятность перехода к новому состоянию равняется exp(-delta / t), где t – текущая температура. Если текущее состояние выгоднее лучшего, то лучшее состояние становится равным новому.

Как только температура становится меньшей, чем заданная минимальная, либо превышен лимит по указанному времени, алгоритм прекращает работу.

### 3.2.3 Сохранение графика результатов

На каждой итерации минимизируемая оценка выгодности состояния равная суммарному расстоянию всех маршрутов состояния сохраняется в специальный контейнер (см. класс ChartData). Т.к. итераций бывает много, хранить оценки всех состояний не оптимально и может вызвать переполнение памяти. Поэтому при достижении лимита количества точек (по умолчанию 1000) график сжимается, последовательно извлекая из каждой тройки соседних точек минимальную и максимальную точку.

### 3.2.4 Визуализация выполнения алгоритма

Модуль интерфейса, получая текущее состояние выполнения отображает график, список загрузок и пройденный путь транспорта, а также граф и анимированный список маршрутов посредством HTML/CSS и JavaScript.

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

Входной файл должен соответствовать следующему формату:

CAPACITY C

DIMENSION N

NODE\_COORD\_SECTION

N1 X1 Y1

N2 X2 Y2

…

Nn Xn Yn

DEMAND\_SECTION

N1 D1

N2 D2

…

Nn Dn

EOF

Где C – ограничение по грузоподъёмности на единицу транспорта, N – количество вершин графа, Ni – номер вершины, Xi – x-координата вершины, y-координата вершины, Di – величина груза вершины.

Выходной файл должен соответствовать следующему формату:

Route #1: v1 v2 … vk

Route #2: v1 v2 … vk

…

Route #M: v1 v2 … vk

cost T

Где M – количество маршрутов, vi – вершина маршрута, T – сумма дистанций всех маршрутов.

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

- Операционная система Windows версии 7 и выше

- Наличие экрана с разрешением не менее 640x480

- Размер оперативной памяти не менее 1 гигабайт

- Наличие 300 мегабайт свободного пространства на жестком диске.

# 4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## 4.1. Ориентировочная экономическая эффективность

В рамках данной работы расчёт экономической эффективности не предусмотрен.

## 4.2. Предполагаемая потребность

Данный продукт должен быть востребован в сфере решения задач маршрутизации, а также для наглядной демонстрации работы алгоритма имитации отжига.

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

На момент начала разработки аналогов разрабатываемого программного обеспечения не нашлось в открытом доступе.

# 5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. A Simulated Annealing Algorithm for The Capacitated Vehicle Routing Problem H. Harmanani, D. Azar, N. Helal W. Keirouz Department of Computer Science & Mathematics Department of Computer Science Lebanese American University American University of Beirut Byblos, 1401 2010, Lebanon Beirut, 1107 2020, Lebanon.
9. Статья об алгоритме имитации отжига для задачи CVRP, Networking and Emerging Optimization [Электронный ресурс] // URL: <http://neo.lcc.uma.es/vrp/solution-methods/metaheuristics/simulated-annealing/>
10. Computer technologies department, ITMO University. Задачи маршрутизации транспорта [Электронный ресурс] // URL: http://rain.ifmo.ru/cat/view.php/theory/unsorted/vrp-2006

# Приложение 1

**Описание и функциональное назначение классов, структур и компонентов**

|  |  |
| --- | --- |
| Класс\Структура\Компонент | Назначение |
| AdvancedTimer | Служит для измерения времени |
| AnnealingSolver | Реализует исполнение алгоритма имитации отжига |
| ChartData | Содержит данные графика |
| GraphData | Содержит данные графа |
| InterfaceGate | Служит для связи с модулем интерфейса |
| RouteSolution | Содержит решение в виде набора маршрутов |
| App.vue | Является точкой запуска интерфейса |
| CarsInfo.vue | Является частью интерфейса со списком единиц транспорта |
| DisplayMode.vue | Является частью интерфейса с кнопкой переключения режима отображения |
| Files.vue | Является частью интерфейса с кнопками загрузки и сохранения файлов |
| Graph.vue | Является частью интерфейса с визуализацией графа |
| ProcessControl.vue | Является частью интерфейса с кнопками запуска, паузы и остановки алгоритма решения |
| Progress.vue | Является частью интерфейса с полосой выполнения алгоритма |
| ResultChart.vue | Является частью интерфейса с визуализацией графика результата |
| Routes.vue | Является частью интерфейса с визуализацией маршрутов |
| Settings.vue | Является частью интерфейса с настройками запуска алгоритма |
| Stats.vue | Является частью интерфейса с отображением параметров статистики |

# Приложение 2

**Описание и функциональное назначение полей, свойств и методов классов, структур и компонентов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс AdvancedTimer** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| duration | private | long long | Значение продолжительности в миллисекундах | |
| active | bool | Флаг работы таймера | |
| last\_point | time\_point | Значение последнего временного момента | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| AdvancedTimer | public | конструктор | - | Конструктор таймера |
| run | void | - | Запускает таймер |
| pause | void | - | Останавливает таймер |
| value | long long | - | Возвращает длительность временного промежутка в миллисекундах |
| clear | void | - | Сбрасывает таймер |
| now | private | time\_point | - | Возвращает текущий момент времени |
| dist | long long | time\_point a, time\_point b | Находит длительность промежутка времени между двумя моментами времени в миллисекундах |
| **Класс AnnealingSolver** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| bestSolution | private | RouteSolution\* | Ссылка на объект лучшего решения | |
| currentSolution | RouteSolution\* | Ссылка на объект текущего решения | |
| chart | ChartData\* | Ссылка на объект данных графика | |
| timer | AdvancedTimer | Таймер для замера времени решения | |
| t | long double | Текущая температура | |
| factor | long long | Коэффициент охлаждения температуры | |
| tEnd | long long | Конечная температура | |
| timeLimit | long long | Лимит по времени в миллисекундах | |
| iterationsTotal | long long | Общее количество итераций | |
| iterationsCnt | long long | Текущее количество итераций | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| AnnealingSolver | public | конструктор | - | Инициализирует решение |
| ~AnnealingSolver | деструктор | - | Очищает память решения |
| isFinished | bool | - | Проверка не завершено ли решение |
| runIteration | void | - | Запустить итерацию |
| pause | public | void | - | Поставить решение на паузу |
| unpause | void | - | Продолжить решение |
| makeUpdate | PyObject\* | - | Упаковать текущее состояние для передачи интерфейсу |
| makeStats | private | PyObject\* | - | Упаковать параметры статистики текущего состояния для передачи интерфейсу |
| makeCars | PyObject\* | RouteSolution\* | Упаковать информацию о транспортных средствах текущего состояния для передачи интерфейсу |
| makeRoutes | PyObject\* | RouteSolution\* | Упаковать информацию о маршрутах текущего состояния для передачи интерфейсу |
| makeChart | PyObject\* | - | Упаковать точки графика результата текущего состояния для передачи интерфейсу |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс ChartData** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| MAX\_LEVEL | private | int | Константа – максимальное количество уровней сжатия | |
| DEFAULT\_MAX\_SIZE | int | Константа – максимальное количество точек графика по умолчанию | |
| max\_size | int | Максимальное количество точек графика | |
| size | int | Текущее количество точек графика | |
| levels | deque <pair <long long, double> >[] | Уровни сжатия, содержащие точки графика | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| ChartData | public | конструктор | int max\_size | Инициализирует контейнер точек графика |
| add\_node | void | long long x, double y | Добавляет точку в контейнер |
| get\_points | vector <pair <long long, double> > | - | Возвращает массив точек графика |
| compress\_data | private | void | - | Сжимает контейнер |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс GraphData** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| nodes | private | Node\* | Массив вершин | |
| n | public | int | Количество вершин | |
| c | int | Ограничение по грузоподъёмности | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| GraphData | public | конструктор | PyObject \*problem | Инициализирует граф |
| ~GraphData | деструктор | - | Удаляет граф из памяти |
| dist | double | int a, int b | Возвращает расстояние между вершинами |
| demand | int | int a | Возвращает нагрузку вершины |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс InterfaceGate** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| bridge\_dict | private | PyObject\* | Словарь функций доступа к мосту интерфейса | |
| state | PyGILState\_STATE | Состояние интерпретатора моста интерфейса | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| pylock | public | void | - | Занять мост интерфейса для обращения |
| pyunlock | void | - | Разблокировать мост интерфейса |
| init | void | - | Инициализировать интерфейс |
| extract\_event | PyObject\* | - | Получить событие интерфейса |
| sendAppEvent | void | string newState, PyObject \*update | Отправить событие в интерфейс |
| call | private | PyObject\* | string func, PyObject \*args | Вызвать функцию моста интерфейса |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс RouteSolution** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| graph | public | GraphData\* | Ссылка на объект графа | |
| routes | vector <Route\*> | Массив маршрутов | |
| cost | double | Оценка решения | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| RouteSolution | public | конструктор | GraphData \*graph | Инициализирует первое случайное решение |
| ~RouteSolution | public | конструктор | RouteSolution \*prev | Инициализирует новое решение путём случайного изменения предыдущего |
| count\_cost | private | void | – | Пересчитывает оценку решения |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Класс RectContent** | | | | | | | | |
| **Поля** | | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Описание | | | |
| paint | private | | Paint | | Краска. | | | |
| rect | Rect | | Границы. | | | |
| **Методы** | | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | | Описание |
| RectContent | public | | статический  конструктор | | – | | | Инициализирует краску. |
| RectContent | констуктор | | Rect rect, Pivot parentPivot, float x, float y | | | Конструктор прямоугольной визуальной единицы. |
| Draw | protected | | void | | Canvas canvas | | | Рисует прямоугольник |
| **Класс DrawablePart** | | | | | | | | |
| **Поля** | | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Описание | | |
| pivot | | public | | Pivot | | Опорная точка | | |
| **Методы** | | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Аргументы | Описание | |
| DrawablePart | | public | | конструктор | | Pivot parentPivot, float x, float y | Конструктор визуальной единицы. | |
| OnDraw | | void | | Canvas canvas | Смещает полотно к опорной точке и вызывает отрисовку. | |
| Draw | | protected | | void | | Canvas canvas | Базовый метод для рисования визуальной единицы. | |
| OnAppearanceChanged | | void | | – | Служит для вызова запроса на перерисовку. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс TopBar** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| Instance | public | TopBar | Хранит экземпляр объекта. | |
| dayNightSwitch | private | SwitchButton | Кнопка переключения яркости. | |
| soundSwitch | SwitchButton | Кнопка переключения звука. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| TopBar | public | конструктор | – | Конструктор верхней панели. |
| BeforeShow | protected | void | – | Активирует внутренние объекты перед показом. |
| BeforeHide | void | – | Деактивирует внутренние объекты перед скрытием. |
| Draw | void | Canvas canvas | Рисует верхнюю панель. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Класс TaskRegistrator** | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| tasks | public | HashSet  <AsyncTask> | Выполняющиеся в фоновом потоке задачи. | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание | |
| CancelAllTasks | public | void | – | Отменяет все задачи. | |
| **Класс Scene** | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| SwitchDuration | private | int | Константа – время перехода между сценами. | | |
| SwitchEasingFactor | float | Константа – коэффициент для интерполятора анимации. | | |
| switchAnim | HandyAnimator | Анимация перехода. | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Описание |
| Scene | public | конструктор | – | | Конструктор сцены |
| Show | void | Side fromSide | | Запускает анимацию появления сцены с запрошенной стороны и действия при появлении. |
| Hide | protected | void | Side toSide | | Запускает анимацию скрытия сцены в запрошенную стороны и действия при скрытии. |
| BeforeShow | void | – | | Базовый метод для действий перед появлением. Добавляет сцену в отрисовку. |
| AfterShow | void | – | | Базовый метод для действий после появления. |
| BeforeHide | void | – | | Базовый метод для действий перед скрытием. |
| AfterHide | void | – | | Базовый метод для действий после скрытия. Убирает сцену из отрисовки. |
| ReleaseAnimation | private | void | – | | Завершает анимацию. |
| AnimateSwitch | void | float fromX, float toX, Action action | | Анимирует переход по абсциссе между заданными значениями. Добавляет к завершению анимации действие. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс Geom** | | | | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| DotProduct | public | long | Vector a, Vector b | Вычисляет величину скалярного произведения векторов. |
| CrossProduct | long | Vector a, Vector b | Вычисляет величину векторного произведения векторов. |
| Dist | double | Point a, Point b | Вычисляет расстояние между точками. |
| Dist | double | Segment s, Point p | Вычисляет расстояние между точкой |
| IsIntersected | bool | Segment a, Segment b | Проверяет пересекаются ли отрезки. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс HLineSwitch** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| paint | private | Paint | Краска. | |
| SwitchDuration | int | Константа – время передвижения линии в миллисекундах. | |
| width | int | Ширина линии. | |
| segmentWidth | int | Ширина подотрезка. | |
| indent | int | Величина отступа. | |
| states | int | Количество состояний. | |
| currentState | int | Текущее состояние. | |
| curX | int | Текущая абсцисса. | |
| anim | HandyAnimator | Анимация передвижения. | |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Описание |
| CurrentState | public | int | get/set | Предоставляет доступ к currentState. При изменении применяет анимацию. |
| CurX | private | int | get/set | Предоставляет доступ к curX. При изменении запрашивает перерисовку. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| HLineSwitch | public | статический  конструктор | – | Инициализирует краску. |
| HLineSwitch | конструктор | int width, int segmentWidth, int states, int currentState, Pivot parentPivot, float x, float y | Конструктор передвигающейся линии. |
| Draw | protected | void | Canvas canvas | Рисует линию. |
| CountStateX | private | int | int state | Высчитывает абсциссу по текущему состоянию. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Класс MainActivity** | | | | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание | |
| OnCreate | protected | void | Bundle bundle | Инициализирует приложение. | |
| OnDestroy | void | – | Деинициализирует приложение. | |
| InitScenes | private | void | – | Создаёт сцены. | |
| RemoveScenes | void | – | Удаляет сцены. | |
| **Класс GameButtonBar** | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| ExitRequested | public | event Action | Событие запроса выхода. | | |
| ReplayRequested | event Action | Событие запроса новой игры. | | |
| lPos | private | float | Константа – левая позиция. | | |
| lmPos | float | Константа – левая средняя позиция. | | |
| mPos | float | Константа – средняя позиция. | | |
| rmPos | float | Константа – правая средняя позиция. | | |
| rPos | float | Константа – правая позиция. | | |
| farMPos | float | Константа – дальняя средняя позиция. | | |
| farRMPos | float | Константа – дальняя правая средняя позиция. | | |
| farRPos | float | Константа – дальняя правая позиция. | | |
| DurationFactor | float | Константа – множитель длительности анимации. | | |
| exitButton | Button | Кнопка выхода. | | |
| cancelButton | Button | Кнопка отмены выхода. | | |
| replayButton | Button | Кнопка новой игры. | | |
| text | TextContent | Текст выхода. | | |
| exitAnim | HandyAnimator | Анимация кнопки выхода. | | |
| sureAnim | HandyAnimator | Анимация текста и кнопки отмены выхода. | | |
| replayAnim | HandyAnimator | Анимация кнопки новой игры. | | |
| state | BarState | Состояние. | | |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Описание | |
| State | private | BarState | get/set | Предоставляет доступ к state. При изменении применяет анимации. | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание | |
| GameButtonBar | public | конструктор | Pivot parentPivot, int x, int y | Конструктор панели игровых кнопок | |
| ResetToNormalState | void | – | Рисует точку. | |
| OnGameEnd | void | – | Показывает предложение новой игры. | |
| Activate | void | – | Активирует кнопки. | |
| Deactivate | void | – | Деактивирует кнопки. | |
| Draw | protected | void | Canvas canvas | Рисует игровую панель кнопок. | |
| KillAnims | private | void | – | Деактивирует анимации. | |
| OnExitButton | void | – | Срабатывает при нажатии выхода. | |
| OnCancelButton | void | – | Срабатывает при нажатии отмены выхода. | |
| OnReplayButton | void | – | Срабатывает при нажатии запроса новой игры. | |
| **Класс TextContent** | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| text | private | string | Текст. | | |
| colorID | int | Номер цвета. | | |
| size | float | Размер текста. | | |
| originX | int | Левая граница. | | |
| originY | int | Нижняя граница. | | |
| paint | Paint | Краска. | | |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | | Описание |
| Text | public | PointState | get/set | | Предоставляет доступ к text. При изменении пересчитывает границы и вызывает перерисовку. |
| ColorID | float | get/set | | Предоставляет доступ к colorID. При изменении вызывает перерисовку. |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Описание |
| TextContent | public | статический  конструктор | – | | Инициализирует краску со шрифтом. |
| TextContent | констуктор | string text, int colorID, float size, Pivot parentPivot, float x, float y | | Конструктор визуальной единицы текста. |
| Draw | protected | void | Canvas canvas | | Рисует текст. |
| RecountOrigin | private | void | – | | Пересчитывает границы. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Структура Geom.Point** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| x | public | int | Абсцисса. | |
| y | int | Ордината. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| Point | public | конструктор | int x, int y | Конструктор точки. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс SwitchButton** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| StateChanged | public | event Action<int> | Событие изменения состояния. | |
| state | private | int | Текущее состояние. | |
| faces | DrawablePart[] | Визуальные единицы при различных состояниях. | |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Описание |
| State | public | int | get/set | Предоставляет доступ к state. При изменении меняет внешний вид, вызывает событие изменения состояния и запрашивает перерисовку. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| SwitchButton | public | конструктор | int defaultState, DrawablePart[] faces, Rect localBounds, Pivot parentPivot, float x, float y | Конструктор меняющейся кнопки. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс HelpScene** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| Instance | public | HelpScene | Экземпляр объекта. | |
| rules | private | string[] | Строки с правилами игры. | |
| backButton | Button | Кнопка назад. | |
| titleText | TextContent | Визуальная единица заголовка. | |
| rulesText | TextContent[] | Визуальные единицы правил. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| GamePoint | public | конструктор | – | Конструктор сцены с правилами. |
| BeforeShow | protected | void | – | Активирует сцену с правилами перед появлением. |
| BeforeHide | void | – | Деактивирует сцену с правилами после скрытия. |
| Draw | void | Canvas canvas | Рисует сцену с правилами. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс Button** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| pressed | public | event Action | Событие при нажатии кнопки. | |
| face | protected | DrawablePart | Визуальная единица кнопки. | |
| DiveScale | private | float | Множитель размера при надавливании. | |
| DiveDuration | int | Время углубления. | |
| disturbed | bool | Надавлена ли кнопка. | |
| diveAnim | HandyAnimator | Анимация углубления. | |
| currentDiveScale | float | Текущий размер при углублении. | |
| currentDiveTime | float | Текущее время углубления. | |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Описание |
| CurrentDiveTime | private | float | get/set | Предоставляет доступ к currentDiveTime. При изменении пересчитывает currentDiveScale и запрашивает перерисовку. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| Button | public | конструктор | Pivot parentPivot, int x, int y | Конструктор игровой точки. |
| OnTouchDown | void | int x, int y | Обрабатывает поведение кнопки при опускании пальца. |
| OnTouchUp | void | int x, int y | Обрабатывает поведение кнопки при поднятии пальца. |
| OnTouchCancel | void | int x, int y | Обрабатывает поведение кнопки при поднятии пальца вне кнопки. |
| Draw | protected | void | Canvas canvas | Рисует кнопку. |
| AnimateDivingTo | private | void | int divingTimeDest | Запускает анимацию размера кнопки в соответствии с желаемой отметкой времени анимации. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Класс GameField** | | | | | | | | |
| **Поля** | | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Описание | | |
| PlayerMoved | | public | | event Action<int> | | Событие при ходе игрока. | | |
| bounds | | private | | Rect | | Границы игрового поля. | | |
| fieldData | | FieldData | | Информация об игровом поле. | | |
| state | | FieldState | | Текущее состояние | | |
| points | | GamePoint[] | | Точки. | | |
| dottedSegment | | GameSegment | | Пунктирный отрезок. | | |
| segments | | List  <GameSegment> | | Проведенные отрезки. | | |
| availableSegments | | SortedSet<int> | | Доступные отрезки. | | |
| pointsAlive | | SortedSet<int> | | Доступные точки. | | |
| pointsTargeted | | SortedSet<int> | | Точки, доступные для хода от точки pointA. | | |
| pointA | | int | | Номер первой выбранной точки. | | |
| pointB | | int | | Номер второй выбранной точки. | | |
| **Методы** | | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Аргументы | Описание | |
| GameField | | public | | статический  конструктор | | – | Инициализирует границы. | |
| GameField | | конструктор | | FieldData fieldData, Pivot parentPivot, float x, float y | Конструктор игрового поля. | |
| AnimateAppearance | | void | | – | Анимирует появление. | |
| ClearEvents | | void | | – | Очищает события. | |
| OnCompetitorsMove | | void | | int segID | Обрабатывает ход соперника. | |
| OnTouchDown | | void | | int x, int y | Обрабатывает жест нажатия. | |
| OnTouchMove | | void | | int x, int y | Обрабатывает жест движения. | |
| OnTouchUp | | void | | int x, int y | Обрабатывает жест отжатия. | |
| OnTouchOutside | | void | | int x, int y | Обрабатывает жест ухода. | |
| OnTouchCancel | | void | | int x, int y | Обрабатывает отмену жеста. | |
| Deactivate | | void | | – | Деактивирует игровое поле. | |
| Draw | | protected | | void | | Canvas canvas | Рисует поле. | |
| AddSegment | | private | | void | | int point1, int point2, int colorID | Добавляет отрезок на поле. | |
| FindAlivePointID | | int | | int x, int y | Ищет доступную точку под пальцем. | |
| ClearTargets | | void | | – | Очищает подсвеченные точки. | |
| MakeTargets | | void | | int pointID | Подсвечивает доступные точки. | |
| PerformSegment | | void | | – | Проводит отрезок игрока между pointA и pointB | |
| GoToFreeStateFromAnyTouchedState | | void | | – | Переходит в состояние без активных точек. | |
| **Класс TouchablePart** | | | | | | | |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | Описание | | |
| localBounds | | protected | | Rect | Границы в локальной системе отсчета. | | |
| bounds | | private | | Rect | Границы в глобальной системе отсчета. | | |
| **Свойства** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | Доступ | | Описание |
| Bounds | | public | | Rect | get | | Рассчитывает глобальные границы. |
| **Методы** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | Аргументы | | Описание |
| TouchablePart | | public | | конструктор | Rect localBounds, Pivot parentPivot, float x, float y | | Конструктор единицы, принимающей сенсорные взаимодействия. |
| OnTouchDown | | void | int x, int y | | Базовый метод для нажатия. |
| OnTouchUp | | void | int x, int y | | Базовый метод для отжатия. |
| OnTouchMove | | void | int x, int y | | Базовый метод для движения. |
| OnTouchOutside | | void | int x, int y | | Базовый метод для ухода. |
| OnTouchCancel | | void | int x, int y | | Базовый метод для отмены. |
| Activate | | void | – | | Базовый метод для активации. Добавляет себя в список активных объектов. |
| Deactivate | | void | – | | Базовый метод для деактивации. Убирает себя из списка активных объектов. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс TouchHandler** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| listeners | public | HashSet  <TouchablePart> | Объекты, которые принимают нажатия сейчас. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| RemoveAllListeners | public | void | – | Очищает все активные объекты. |
| Handle | bool | MotionEvent e | Принимает пользовательский жест. |
| PerformTouchDown | private | void | int x, int y | Обрабатывает жест нажатия. |
| PerformTouchUp | void | int x, int y | Обрабатывает жест отжатия. |
| PerformTouchCancel | void | int x, int y | Обрабатывает отмену жеста. |
| PerformTouchMove | void | int x, int y | Обрабатывает жест движения. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс LogoScene** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| Instance | public | LogoScene | Экземпляр объекта. | |
| fcsLogo | private | BitmapContent | Логотип ФКН. | |
| hseLogo | BitmapContent | Логотип ВШЭ. | |
| supbyText | TextContent | Текст заголовка. | |
| fcsText | TextContent | Текст ФКН. | |
| hseText | TextContent | Текст ВШЭ. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| LogoScene | public | конструктор | – | Конструктор cцены с логотипами. |
| AfterShow | protected | void | – | После появления назначает переход к следующей сцене. |
| Draw | void | Canvas canvas | Рисует сцену с логотипами. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс MenuScene** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| Instance | public | MenuScene | Экземпляр объекта. | |
| titleText | private | TextContent | Текст заголовка. | |
| playButton | Button | Кнопка играть. | |
| helpButton | Button | Кнопка показать правила. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| MenuScene | public | конструктор | – | Конструктор сцены меню. |
| BeforeShow | protected | void | – | Активирует сцену перед появлением. |
| BeforeHide | void | – | Деактивирует сцену перед скрытием. |
| Draw | void | Canvas canvas | Рисует сцену меню. |
| CreateHexagonButton | private | Button | int size, int id, float x, float y | Создаёт кнопку в форме шестиугольника. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс SingleGameScene** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| Instance | public | SingleGameScene | Экземпляр объекта. | |
| titleText | private | TextContent | Текст заголовка. | |
| statusText | TextContent | Текст текущего статуса игры. | |
| bar | GameButtonBar | Панель с игровыми кнопками. | |
| field | GameField | Игровое поле. | |
| logic | SingleGameLogic | Логика игры. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| SingleGameScene | public | конструктор | – | Конструктор сцены игры. |
| OnGameStatusChanged | void | SingleGameLogic.GameStatus status | Обрабатывает событие изменения статуса игры. |
| InitGame | void | SingleGameLogic logic | Инициализирует новую игру. |
| BeforeShow | protected | void | – | Перед появлением активирует сцену и сбрасывает её состояние. |
| AfterShow | void | – | После появления анимирует появление игрового поля и запускает игру. |
| BeforeHide | void | – | Перед скрытием деактивирует сцену и останавливает игру. |
| AfterHide | void | – | После скрытия освобождает память, занятую игровыми ресурсами. |
| Draw | void | Canvas canvas | Рисует сцену игры. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Структура Geom.Segment** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| a | public | Point | Первая точка. | |
| b | Point | Вторая точка. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| Segment | public | конструктор | Point a, Point b | Конструктор отрезка. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс SingleGameRunUpScene** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| Instance | public | SingleGameRunUpScene | Экземпляр объекта. | |
| loading | private | LoadingIndicator | Индикатор загрузки. | |
| processingText | TextContent | Текст информирования о загрузке. | |
| easyButton | Button | Кнопка легкой игры. | |
| normalButton | Button | Кнопка нормальной игры. | |
| hardButton | Button | Кнопка сложной игры. | |
| backButton | Button | Кнопка назад. | |
| startButton | Button | Кнопка начать игру. | |
| lineSwitch | HLineSwitch | Передвигающаяся линия, отображающая текущую выбранную сложность. | |
| task | GenerateSingleGameLogicTask | Задача генерации игровой логики. | |
| logic | SingleGameLogic | Игровая логика. | |
| state | SceneState | Текущее состояние сцены. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| SingleGameRunUpScene | public | конструктор | – | Конструктор сцены подготовки к игре. |
| BeforeShow | protected | void | – | Перед показом активирует сцену, включает анимацию загрузки и запускает задачу генерации поля. |
| BeforeHide | void | – | Перед скрытием деактивирует сцену и отменяет задачу генерации, если таковая выполняется. |
| AfterHide | void | – | После скрытия переходит к пустому состоянию сцены. |
| Draw | void | Canvas canvas | Рисует точку. |
| CreateDifficultyButton | private | Button | int res\_id, string text, float x, float y | Создает кнопку сложности. |
| OnTaskFinished | void | SingleGameLogic logic | Запускается по завершению задачи генерации, сохраняет игровое поле, убирает анимацию загрузки и показывает кнопку начать игру. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс GenerateSingleGameLogicTask** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| logic | private | SingleGameLogic | Сгенерированная игровая логика. | |
| callback | Action  <SingleGameLogic> | Функция обратного вызова. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| GenerateSingleGameLogicTask | public | конструктор | – | Конструктор задачи генерации игровой логики. Сохраняет обратный вызов. |
| DoInBackground | protected | Java.Lang.Object | params Java.Lang.Object[] @params | Выполняет создание игровой логике в фоновом потоке. |
| OnPreExecute | void | – | Перед запуском регистрирует задачу в TaskRegistrator. |
| OnCancelled | void | – | При отмене удаляет задачу из TaskRegistrator. |
| OnPostExecute | void | Java.Lang.Object result | При завершении удаляет задачу из TaskRegistrator и запускает обратный вызов. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Структура Bitmask128** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| Zero | public | Bitmask128 | Константа – нулевая маска. | |
| l | ulong | Левая часть маски. | |
| r | ulong | Правая часть маски. | |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Описание |
| this[int i] | public | bool | get/set | Предоставляет доступ к битам. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| Bitmask128 | public | конструктор | ulong l, ulong r | Конструктор битовой маски. |
| Bitmask128 | конструктор | int firstTrueBitsCnt | Конструктор битовой маски. |
| operator & | Bitmask128 | Bitmask128 a, Bitmask128 b | Применяет к двум маскам логическую операцию И. |
| GetHashCode | int | – | Возвращает хэш-код. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс FieldDataGenerator** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| random | private | Random | Генератор случайных значений. | |
| Width | int | Универсальная дискретная ширина поля. | |
| Height | int | Универсальная дискретная высота поля. | |
| N | int | Количество точек. | |
| MinPointDist | int | Минимальное расстояние между точками. | |
| Iterations | int | Количество итераций алгоритма отжига. | |
| StartTemperature | double | Начальная температура алгоритма отжига. | |
| TemperatureMultiplier | double | Множитель температуры алгоритма отжига. | |
| points | Geom.Point[] | Геометрическое место точек. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| Generate | public | FieldData | – | Генерирует точки запуская алгоритм отжига. |
| CountError | private | double | – | Считает текущую оценочную функцию для алгоритма отжига. |
| RandomPoint | Geom.Point | – | Генерирует случайную точку. |
| Anneal | double | – | Алгоритм отжига. После завершения возвращает значение глобального минимума. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Класс FieldData** | | | | | | | |
| **Поля** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Описание | | |
| InvalidSegmentID | public | | int | | Неправильный номер отрезка. | | |
| pointsCnt | int | | Количество точек. | | |
| segmentsCnt | int | | Количество отрезков. | | |
| points | Geom.Point[] | | Геометрическая информация о точках. | | |
| segments | Geom.Segment[] | | Геометрическая информация об отрезках. | | |
| segmentID | int[,] | | Номер отрезка по номерам точек. | | |
| intersectedWith | int[][] | | Список отрезков, с которыми пересекается отрезок. | | |
| pointAbySegment | int[] | | Первая точка отрезка. | | |
| pointBbySegment | int[] | | Вторая точка отрезка. | | |
| MinSegmentDist | private | | int | | Минимальное расстояние между отрезком и точкой. | | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Описание |
| FieldData | public | | конструктор | | Geom.Point[] points | | Конструктор информации об игровом поле. |
| **Класс ColorBank** | | | | | | | |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | Описание | | |
| colors | | private | | Color[] | Массив цветов. | | |
| Black | | int | Номер черного цвета. | | |
| White | | int | Номер белого цвета. | | |
| Blue | | int | Номер синего цвета. | | |
| Red | | int | Номер красного цвета. | | |
| Yellow | | int | Номер желтого цвета. | | |
| Background | | int | Номер фонового цвета. | | |
| BgAnimDuration | | int | Длительность анимации фона в миллисекундах. | | |
| currentBgCoef | | float | Текущий коэффициент фона. | | |
| bgAnim | | HandyAnimator | Анимация фона. | | |
| **Свойства** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | Доступ | Описание | |
| CurrentBgCoef | | public | | float | get/set | Предоставляет доступ к currentBgCoef. При изменении пересчитывает фоновый цвет и запрашивает перерисовку. | |
| **Методы** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | Аргументы | Описание | |
| ColorBank | | public | | статический конструктор | – | Инициализирует фоновый цвет из сохраненных настроек. | |
| ChangeBackgroundColor | | void | bool toBlack | Анимирует переход фонового цвета по запросу. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс Pivot** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| Changed | public | event Action | События изменения позиции. | |
| \_pivot | private | Pivot | Опорная точка – родитель. | |
| \_x | float | Абсцисса. | |
| \_y | float | Ордината. | |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Описание |
| Parent | public | Pivot | get/set | Предоставляет доступ к \_pivot. При изменении вызывает Changed. |
| X | float | get/set | Предоставляет доступ к \_x. При изменении вызывает Changed. |
| Y | float | get/set | Предоставляет доступ к \_y. При изменении вызывает Changed. |
| AbsX | float | get | Вычисляет абсолютную абсциссу. |
| AbsY | float | get | Вычисляет абсолютную ординату. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| Pivot | public | конструктор | float x, float y, Pivot parent | Конструктор опорной точки. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс SoundMaster** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| ButtonSound | public | int | Номер звука кнопки. | |
| FailureSound | int | Номер звука проигрыша. | |
| PointBornSound | int | Номер звука появления кнопки. | |
| PointChosenSound | int | Номер звука подсветки кнопки. | |
| SegmentBornSound | int | Номер звука появления отрезка. | |
| VictorySound | int | Номер звука победы. | |
| SceneSwitchSound | int | Номер звука перехода сцены. | |
| MaxStreams | private | int | Максимальное количество звуковых потоков. | |
| soundPool | SoundPool | Объект звукового пула. | |
| lastStream | int | Номер последнего звукового потока. | |
| volume | float | Громкость. | |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Описание |
| Volume | public | float | get/private set | Предоставляет доступ к volume. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| SoundMaster | public | статический конструктор | – | Инициализирует звуковой пул и загружает настройки громкости. |
| SetVolume | void | float volume | Устанавливает громкость. |
| LoadSounds | void | – | Загружает все звуки. |
| PlaySound | void | int soundID | Играет запрошенный звук. |
| StopAllSounds | void | – | Останавливает все звуки. |
| UnloadSounds | void | – | Выгружает все звуки. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс SingleGameLogic** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| fieldData | public | FieldData | Информация об игровом поле. | |
| StatusChanged | event Action  <GameStatus> | Событие изменение статуса игры. | |
| BotMoved | event Action<int> | Событие хода искусственного интеллекта. | |
| StartGameDelay | private | int | Константа – задержка перед началом игры в миллисекундах. | |
| BotThinkingDuration | int | Константа – время задержки хода бота в миллисекундах. | |
| BotMovingDuration | int | Константа – время хода бота в миллисекундах. | |
| MaxBotDepth | int | Константа – максимальная глубина видимости для бота. | |
| statePredictions | Dictionary  <Bitmask128, float[]> | Оценки выгодностей исходов с различных глубин. | |
| curStateMask | Bitmask128 | Текущий исход игры. | |
| segmentProhibitMask | Bitmask128[] | Маски запрета для отрезков. | |
| botDepth | int | Установленная величина глубины для бота. | |
| random | Random | Генератор случайных значений. | |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Описание |
| BotDepth | public | int | get/set | Предоставляет доступ к botDepth. При изменении сохраняет настройки. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| SingleGameLogic | public | статический конструктор | – | Загружает настройки глубины для бота. |
| SingleGameLogic | конструктор | – | Конструктор игровой логики. |
| OnPlayersMove | void | int segID | Обрабатывает ход игрока. |
| StartGame | void | – | Начинает игру. |
| ClearEvents | void | – | Очищает события. |
| MakeStatePredictions | private | float[] | Bitmask128 mask | Просчитывает выгодности состояний для данного исхода. |
| DelayAction | void | Action action, int duration | Назначает задержку входному действию. |
| MakeBotMove | void | – | Делает ход бота. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс HandyAnimator** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| Update | public | event Action  <Java.Lang.Object> | Событие обновления анимации. | |
| After | event Action | Событие завершения анимации. | |
| core | ValueAnimator | Анимация. | |
| animators | private | HashSet  <HandyAnimator> | Все активные анимации. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| OnActivityDestroy | public | void | – | По завершению активности приложения удаляет все действия по завершению анимации и останавливает анимации. |
| RemoveAfterActions | void | – | Удаляет действие по завершению анимации. |
| OfFloat | HandyAnimator | float from, float to, int duration | Создает анимацию вещественного числа. |
| OfArgb | HandyAnimator | int from, int to, int duration | Создает анимацию цвета. |
| OfNothing | HandyAnimator | int duration | Создает анимацию - задержку. |
| HandyAnimator | private | конструктор | ValueAnimator coreAnim, int duration | Конструктор анимации. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Структура Geom.Vector** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| x | public | int | Направление по абсциссе. | |
| y | int | Направление по ординате. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| Vector | public | конструктор | int x, int y | Конструктор вектора. |
| Vector | конструктор | Point a, Point b | Конструктор вектора. |
| Length | double | – | Вычисляет длину вектора. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс GameView** | | | | |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Описание | |
| CanonWidth | public | int | Константа – каноническая ширина экрана. | |
| CanonHeight | int | Константа – каноническая высота экрана. | |
| xCenter | int | Центр экрана по абсциссе. | |
| yCenter | int | Центр экрана по ординате. | |
| scaleFactor | float | Коэффициент отношения текущего размера экрана к каноническому. | |
| DrawEvent | event Action  <Canvas> | Событие отрисовки. | |
| rootPivot | Pivot | Корневая опорная точка. | |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Описание |
| Instance | public | GameView | get/set | Экземпляр объекта. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| GameView | public | статический конструктор | – | Рассчитывает данные о размерах экрана. |
| GameView | конструктор | Context context, IAttributeSet attrs | Конструктор элемента интерфейса для отрисовки. |
| OnTouchEvent | bool | MotionEvent e | Передает обработку события жеста классу TouchHandler. |
| OnDraw | protected | void | Canvas canvas | Рисует фон, а затем всё остальное. |

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| изменённых | заменённых | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |