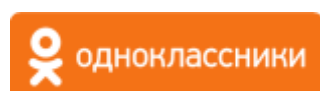




# CODETANKS 2012

## ПРАВИЛА



Октябрь, 2012

# Оглавление

<b>1</b>	<b>Объявление о проведении Конкурса</b>	<b>2</b>
1.1	Наименование Конкурса	2
1.2	Информация об организаторе конкурса	2
1.3	Сроки проведения Конкурса	3
1.4	Условие получения статуса Участника конкурса	3
1.5	Срок регистрации Участников конкурса в Системе Организатора	3
1.6	Территория проведения Конкурса	3
1.7	Условия проведения Конкурса (существо заданий, критерии и порядок оценки)	3
1.8	Порядок определения Победителей и вручения Призов. Призовой фонд Конкурса	4
1.9	Порядок и способ информирования участников Конкурса	5
<b>2</b>	<b>О мире CodeTanks 2012</b>	<b>6</b>
2.1	Общие положения и правила	6
2.2	Описание игрового мира	7
2.3	Описание танков	9
2.4	Описание снарядов и механики попаданий	10
2.5	Описание бонусов	11
2.6	Начисление баллов	11
<b>3</b>	<b>Создание стратегии</b>	<b>12</b>
3.1	Техническая часть	12
3.2	Управление танком	13
3.3	Примеры реализации	15
3.3.1	Пример для Java	15
3.3.2	Пример для C#	15
3.3.3	Пример для C++	16
3.3.4	Пример для Python 2 и Python 3	16
3.3.5	Пример для Pascal	16
<b>4</b>	<b>Package model</b>	<b>18</b>
4.1	Classes	19
4.1.1	CLASS <b>Bonus</b>	19
4.1.2	CLASS <b>BonusType</b>	19
4.1.3	CLASS <b>FireType</b>	20
4.1.4	CLASS <b>Move</b>	21
4.1.5	CLASS <b>Obstacle</b>	22
4.1.6	CLASS <b>Player</b>	22
4.1.7	CLASS <b>Shell</b>	23
4.1.8	CLASS <b>ShellType</b>	23
4.1.9	CLASS <b>Tank</b>	24
4.1.10	CLASS <b>TankType</b>	27
4.1.11	CLASS <b>Unit</b>	27
4.1.12	CLASS <b>World</b>	29

<b>5</b>	<b>Package &lt;none&gt;</b>	<b>31</b>
5.1	Interfaces . . . . .	32
5.1.1	INTERFACE <b>Strategy</b> . . . . .	32

# Глава 1

## Объявление о проведении Конкурса

Общество с ограниченной ответственностью «Одноклассники», созданное и действующее в соответствии с законодательством Российской Федерации, с местом нахождения по адресу: 125167, г. Москва, Ленинградский проспект, 47 стр. 2, далее по тексту «Организатор конкурса», приглашает физических лиц, достигших к моменту опубликования настоящего Объявления о конкурсе 18 лет, далее по тексту «Участник конкурса», к участию в конкурсе на нижеследующих условиях:

### 1.1 Наименование Конкурса

«Российский кубок по программированию искусственного интеллекта (Russian AI Cup)».

Целями проведения Конкурса являются:

- повышение общественного интереса к сфере создания программных продуктов;
- предоставление Участникам конкурса возможности раскрыть творческие способности;
- развитие профессиональных навыков Участников конкурса.

Конкурс состоит из 3 (трех) этапов, каждый из которых завершается определением Победителей. Последний этап Конкурса является решающим для Участников конкурса в состязании за получение звания Победителя Конкурса занявшего соответствующее призовое место.

### 1.2 Информация об организаторе конкурса

Наименование - ООО «Одноклассники»;

Адрес места нахождения: 125167, г. Москва, Ленинградский проспект, 47 стр. 2

Почтовый адрес: 125167, г. Москва, Ленинградский проспект, 47 стр. 2

Телефон: (495) 725-63-57/59

Сайт: <http://www.russianaicup.ru>

Е-мейл: [support@russianaicup.ru](mailto:support@russianaicup.ru)

### **1.3 Сроки проведения Конкурса**

Срок проведения Конкурса: с 00.00 часов 29 октября 2012 года до 24.00 часов 2 декабря 2012 года по Московскому времени.

Сроки начала и окончания этапов Конкурса:

- первый этап – с 00 часов 00 минут 10 ноября 2012 года до 24 часов 00 минут 11 ноября 2012 года;
- второй этап – с 00 часов 00 минут 17 ноября 2012 года до 24 часов 00 минут 18 ноября 2012 года;
- третий этап (заключительный) – с 00 часов 00 минут 24 ноября 2012 года до 24 часов 00 минут 25 ноября 2012 года;

### **1.4 Условие получения статуса Участника конкурса**

Для участия в Конкурсе необходимо пройти процедуру регистрации в Системе Организатора конкурса размещенной на сайте Организатора конкурса в сети Интернет по адресу:  
<http://www.russianaicup.ru>.

### **1.5 Срок регистрации Участников конкурса в Системе Организатора**

Регистрация Участников конкурса проводится с 00.00 часов 29 октября 2012 года до 24.00 часов 2 декабря 2012 года включительно.

### **1.6 Территория проведения Конкурса**

Конкурс проводится на территории Российской Федерации. Проведение всех этапов Конкурса осуществляется путем удаленного доступа к Системе Организатора конкурса через сеть Интернет.

### **1.7 Условия проведения Конкурса (существо заданий, критерии и порядок оценки)**

Порядок проведения Конкурса, существо задания, критерии и порядок оценки указаны в конкурсной документации в разделе 2.1.

Конкурсная документация включает в себя:

- Объявление о проведении Конкурса;
- Соглашение об организации и порядке проведения Конкурса;
- Правила проведения Конкурса;
- информационные данные, содержащиеся в Системе Организатора конкурса.

Участник конкурса может ознакомиться с конкурсной документацией на сайте Организатора конкурса в сети Интернет по адресу: <http://www.russianaicup.ru>, а также при прохождении процедуры регистрации в Системе Организатора конкурса.

Организатор конкурса оставляет за собой право на изменение конкурсной документации, условий проведения Конкурса и отказ от его проведения в соответствии с условиями конкурсной документации и нормами законодательства РФ. При этом, Организатор Конкурса обязуется уведомить Участников конкурса о всех произошедших изменениях, путем отправки уведомления, в порядке и на условиях, предусмотренных в конкурсной документации.

## **1.8 Порядок определения Победителей и вручения Призов. Призовой фонд Конкурса**

Критерии оценки результатов Конкурса, количество и порядок определения Победителей содержатся в разделе 2.1 данного документа.

Призовой фонд Конкурса формируется за счет средств Организатора конкурса.

Призовой фонд:

- 1-е место: Apple Macbook Pro 15"Retina
- 2-3 места: Apple Macbook Air 13"
- 4-6 места: Apple iPad New 64Gb Wi-Fi + Cellular
- 1-6 места в квалификации (Песочница): Apple iPod Touch 32 GB

Все участники Конкурса, принявшие участие во втором этапе, будут вознаграждены футболкой.

Все участники занявшие призовые места, будут оповещены посредством отправки сообщения на адрес электронной почты, указанный участником при регистрации в Системе Организатора.

Призы будут высланы участникам в виде посылок, используя Почту России, в течение двух недель после окончания финального этапа. Срок доставки приза по почтовому адресу, указанному участником, зависит от сроков доставки Почты России. Почтовые адреса призеров, для отправки призов, Организатор получает из учетных данных участника в Системе Организатора. Адрес должен быть указан участником-призером в течение трех дней, после получения уведомления о получении приза.

При отсутствии ответа в обозначенные сроки или отказе предоставить точные данные, необходимые для вручения призов Конкурса, Организатор оставляет за собой право отказать такому участнику в выдаче приза Конкурса. Денежный эквивалент приза не выдается

Победители Конкурса обязуются предоставить Организатору конкурса копии всех документов, необходимых для бухгалтерской и налоговой отчетности Организатора конкурса. Перечень документов, которые Победитель обязан предоставить Организатору конкурса, включает в себя:

- копию паспорта Победителя;
- копию свидетельства о постановки на налоговый учет Победителя;
- копию пенсионного удостоверения Победителя;
- данные об открытии банковского лицевого счета Победителя;
- иные документы, которые Организатор конкурса потребует от Участника конкурса в целях формирования отчетности о проведенном Конкурсе.

Наряду с копиями Организатор конкурса вправе запросить оригиналы вышеуказанных документов.

В соответствии с подпунктом 4 пункта 1 статьи 228 НК РФ Победитель Конкурса, ставший обладателем Приза, самостоятельно несет все расходы по уплате всех применимых налогов, установленных действующим законодательством Российской Федерации.

## **1.9 Порядок и способ информирования участников Конкурса**

Информирование Участников Конкурса осуществляется путем размещения информации в сети Интернет на Сайте Организатора конкурса по адресу: <http://www.russianaicup.ru>, а также через Систему Организатора конкурса, в течение всего срока проведения Конкурса.

## Глава 2

# О мире CodeTanks 2012

### 2.1 Общие положения и правила

Данное соревнование предоставляет вам возможность проверить свои навыки программирования, создав искусственный интеллект (стратегию), управляющий одним или взводом танков в специальном игровом мире (подробнее об особенностях мира CodeTanks 2012 можно узнать в следующих разделах). Ваш танк или взвод (численностью до 3 боевых единиц) должен сражаться на поле боя против боевых подразделений (той же численности) нескольких других игроков в режиме «каждый сам за себя (КСЗС)». Победителем считается игрок, набравший максимальное количество баллов. Система подсчета баллов будет описана позднее.

Турнир проводится в несколько этапов, которым предшествует квалификация в Песочнице. Песочница — соревнование, которое проходит на протяжении всего чемпионата. В рамках каждого этапа игроку соответствует некоторое значение рейтинга — показателя того, насколько успешно его стратегия участвует в боях. Начальное значение рейтинга на каждом этапе равно 1200. По итогам боя это значение может как увеличиться, так и уменьшиться. При этом победа над слабым (с низким рейтингом) противником даёт небольшой прирост, также и поражение от сильного соперника незначительно уменьшает ваш рейтинг.

Сначала все участники могут участвовать только в боях, проходящих в Песочнице. Игроки могут отправлять в Песочницу свои стратегии, и последняя принятая из них берётся системой для участия в квалификационных боях. Квалификационные бои проводятся примерно 1 раз в час. Жюри оставляет за собой право изменить этот интервал исходя из пропускной способности тестирующей системы, однако плотность (количество в единицу времени) боёв для всех игроков остаётся постоянной величиной. Бои в Песочнице проходят в формате <sup>1</sup>, соответствующем формату случайного прошедшего этапа турнира или же формату следующего (текущего) этапа. При этом, чем ближе значение рейтинга двух игроков в рамках Песочницы, тем больше вероятность того, что они окажутся в одном бою. Песочница стартует до начала первого этапа турнира и завершается через некоторое время после финального (смотрите расписание этапов для уточнения подробностей). Помимо этого Песочница замораживается на время проведения этапов турнира. По итогам боёв в Песочнице происходит отбор для участия в Раунде 1, в который пройдут 900 участников с наибольшим рейтингом (при его равенстве приоритет отдаётся игроку, раньше отправившему последнюю версию своей стратегии).

Этапы турнира:

- Раунд 1 проверит ваши навыки управления одним танком, который будет заброшен в «клетку»

---

<sup>1</sup>Под форматом боя понимается совокупность таких параметров, как количество игроков, участвующих в нём, и количество танков в команде каждого игрока. Кратко формат записывается в виде `<количество_игроков> × <количество_танков>`, например запись `6 × 1` означает формат боя, в котором участвует 6 игроков, управляющих по одному танку каждый.



вместе с танками пяти других игроков ( $6 \times 1$ ). Несмотря на кажущуюся простоту (ввиду отсутствия необходимости координировать действия с союзными танками), это не будет лёгким испытанием и победу вам буквально придётся вырывать зубами. Этот этап, как и все последующие, состоит из двух частей, между которыми будет небольшой перерыв (с возобновлением работы Песочницы), который позволит улучшить свою стратегию. Для боёв в каждой части выбирается последняя стратегия, отправленная игроком до начала части. Бои проводятся волнами. В каждой волне каждый игрок участвует ровно в одном бою. Количество волн в каждой части определяется возможностями тестирующей системы, но гарантируется, что оно не будет меньше 10. 300 участников с наиболее высоким рейтингом пройдут в Раунд 2.

- В Раунде 2 в каждом бою будет участвовать по 3 игрока, выставляющих небольшие взводы из 2 танков (тип боев  $3 \times 2$ ). Между этапами будет некоторый перерыв, так что у вас будет возможность немного доработать стратегию, добавить координацию между танками. Этот этап будет довольно коварным, так как достаточно двум другим игрокам выбрать ваши танки в качестве приоритетной цели, и никакая хитрая стратегия вас уже не спасёт. Усложняет задачу ещё и то, что после подведения итогов Раунда 1 часть слабых стратегий будет отсеяна и вам придётся противостоять более сильным соперникам. По итогам Раунда 2 лучшие 50 стратегий попадут в Финал.
- Финал является самым серьёзным этапом. После отбора, проведённого по итогам двух первых этапов, останутся сильнейшие. И в каждом бою вам придётся сойтись лицом к лицу с одним из них. У игрока в распоряжении находится взвод из 3 боевых машин (тип боев  $2 \times 3$ ), и здесь взаимодействие танка с товарищами может оказаться решающим фактором. Одна ошибка — и вся ваша команда будет уничтожена. Система проведения Финала имеет свои особенности. Этап по прежнему делится на две части, однако они уже не будут состоять из волн. В каждой части этапа будут проведены дуэли между всеми парами участников Финала. Помимо всего этого будет ещё одно новшество: в боях Финала в центре поля будет находиться бункер — прямоугольный неподвижный объект. Точные размеры и свойства бункера не раскрываются. Вы можете изучить их самостоятельно во время подготовки к Финалу в Песочнице.

После окончания Финала все финалисты упорядочиваются по невозрастанию рейтинга. При равенстве рейтингов более высокое место занимает тот финалист, чья участвовавшая в финале стратегия была отослана раньше. Призы за Финал распределяются на основании занятого места после этого упорядочивания. Лучшие шесть финалистов награждаются призами:

- 1 место — Apple MacBook Pro 15"(Retina),
- 2-3 места — Apple MacBook Air 13",
- 4-6 места — Apple Ipad New (Wi-Fi, Cellular 64Gb).

После окончания Песочницы все её участники упорядочиваются по невозрастанию рейтинга. При равенстве рейтингов более высокое место занимает тот участник, который раньше отослал последнюю версию своей стратегии. Призы за Песочницу распределяются на основании занятого места после этого упорядочивания. Лучшие шесть участников Песочницы награждаются ценными подарками.

На выбор игроку дано 3 типа танков, различающихся по целому ряду параметров: средний танк (СТ), тяжёлый танк (ТТ) и самоходное противотанковое орудие (ПТ) — истребитель танков. Каждая стратегия во время инициализации выбирает себе тип танка, которым и будет управлять во время боя, причём в составе одного взвода могут быть танки разных типов. При создании стратегии игроку следует учесть, что танки разных типов требуют разного стиля ведения боя.

## 2.2 Описание игрового мира

Игровой мир (военный полигон) представляет собой некоторую двумерную прямоугольную область, ограниченную нерушимыми стенами. Размер области —  $1280 \times 800$  точек. Ось абсцисс в этом мире

направлена слева направо, ось ординат — сверху вниз, а положительный угол вращения означает вращение по часовой стрелке. Боевые подразделения игроков расположены на окружности, центр которой совпадает с центром игрового мира. Расстояния между любыми двумя соседними (по или против хода часовой стрелки) взводами танков равны. Изначально все танки направлены в центр игрового мира.

Время в игре дискретное и измеряется в «тиках». В начале каждого «тика» игра получает от стратегий желаемые действия танков в этот тик и обновляет состояние танков в соответствии с этими желаниями и ограничениями мира. Затем происходит расчёт изменения мира и объектов в нём за этот тик, и процесс повторяется снова с обновлёнными данными. Максимальная длительность каждой игры — 5000 тиков. Существуют также случаи, в которых игра может закончиться раньше этого времени, например, в случае, если не уничтоженными остались только танки одного игрока, либо стратегии всех игроков «упали». «Упавшая» стратегия больше не может управлять танками.

Стратегия считается «упавшей» в одном из двух случаев:

- Процесс, в котором запущена стратегия, непредвиденно завершился, либо произошла ошибка в протоколе взаимодействия между стратегией и игровым сервером;
- Стратегия превысила одно (любое) из отведённых ей ограничений по времени. Стратегии на один тик выделяется не более 2 секунд реального времени. Но в сумме за всю игру стратегии выделяется  $50 \times \langle \text{длительность игры в тиках} \rangle$  миллисекунд реального времени и  $20 \times \langle \text{длительность игры в тиках} \rangle$  миллисекунд процессорного времени.<sup>1</sup>

В мире CodeTanks 2012 существует 3 типа объектов. Далее приведено описание их физических и геометрических параметров:

- Танк — объект прямоугольной формы, управляемый игроком (подробнее управление танком будет описано далее). У танка есть башня с орудием, угол поворота которой может отличаться от угла поворота корпуса танка. Ни башня, ни орудие никаким образом не взаимодействуют с другими объектами, таким образом танки могут подъехать вплотную, даже если их орудия направлены друг на друга. На движение танка могут воздействовать внешние факторы, такие, как соударение с другими объектами, сила трения, сила сопротивления воздуха. Скорость танка при столкновении с другим объектом изменяется естественным образом в зависимости от начальных скоростей танка и объекта и их масс. При столкновении двух танков на большой скорости оба они получают урон. Точная формула величины урона не раскрывается, однако известно, что урон зависит от скоростей танков и их масс. Для расчётов берутся длины проекций векторов скорости каждого танка на линию, соединяющую центры этих танков. Проекция вектора скорости, направленная к центру другого танка, усиливает урон, от него — ослабляет. Дополнительно известно, что при столкновении двух танков с одинаковой массой тот танк, у которого длина проекции вектора скорости больше, нанесёт чуть больший урон. Таким образом имеет смысл «таранить» неподвижные или медленно движущиеся танки противника.
- Снаряд — небольшой объект прямоугольной формы, появляющийся при выстреле из танка. Центр его совпадает с концом орудия танка, а направление — с направлением башни/орудия танка. Снаряд имеет некоторую начальную скорость, зависящую от типа снаряда, и далее стратегией никак не управляется. При попадании в танк, снаряд либо взрывается, нанося урон, либо, в зависимости от угла соударения, ricochetит от танка и продолжает лететь дальше. В любом случае скорость как снаряда, так и танка после соударения изменяется в зависимости от их масс. При попадании в стену, при падении на землю (вследствие сильного уменьшения скорости) либо при столкновении с другим снарядом (такое тоже возможно) снаряд взрывается, не нанося никому урона.
- Бонус (полезный предмет) — статический объект квадратной формы, появляющийся случайным образом в различных точках мира. При попадании снаряда в бонус, оба они уничтожаются и исчезают из мира. Танк может подобрать бонус, коснувшись его. При этом скорость танка изменяется так, как если бы он столкнулся с бонусом.

<sup>1</sup> Несмотря на то, что ограничение реального времени заметно выше ограничения процессорного времени, запрещено искусственно «замедлять» тестирование стратегии командами типа «sleep». В случае выявления подобных злоупотреблений, жюри оставляет за собой право применить к данному пользователю меры на своё усмотрение, вплоть до блокировки аккаунта.

## 2.3 Описание танков

Как уже было сказано выше, в игре существует три типа танков.

В целом можно сказать, что СТ — небольшой лёгкий и мобильный танк, основным преимуществом которого являются не броня или высокая прочность корпуса, а способность уходить от атак, заходить во фланг противнику, а также быстро собирать бонусы. В дополнение к этому СТ может двигаться задним ходом ненамного медленнее, чем вперёд.

ТТ имеет чуть бóльшие размеры и гораздо массивнее, его можно назвать достаточно сбалансированным и универсальным. Он менее подвижен, чем СТ, однако нельзя сказать, что он совсем неповоротлив. Главным преимуществом ТТ является повышенная прочность корпуса и достойное лобовое и бортовое бронирование.

ПТ в большинстве случаев представляет собой этакую стационарную турель — боевую единицу поддержки. ПТ имеет высокую прочность корпуса и очень толстую лобовую броню, что вкупе с высокой скоростью перезарядки делает самоубийственной перестрелку с ним лоб в лоб. Однако бортовая броня даже тоньше, чем у среднего танка, а ограничение угла поворота орудия не позволяет стрелять по противникам, если только они не находятся в узком секторе перед ПТ. Вдобавок к выше сказанному, ПТ не слишком мобильно. Всё это делает его очень уязвимым ко фланговым обходам, а также практически исключает возможность сбора бонусов.

В следующей таблице приведены основные характеристики танков. В процессе создания стратегии вы также можете самостоятельно изучать мир и другие параметры различных типов танков и других игровых объектов.

Характеристика танка	Средний танк	Тяжёлый танк	Противотанковое орудие
Ширина, точек	90	105	112.5
Высота, точек	60	75	67.5
Виртуальная длина орудия, точек	67.5	82.5	97.5
Масса танка, единиц массы	10	30	20
Мощность двигателя, единиц мощности	7500	13500	7500
Коэфф. мощности при движении назад	0.75	0.65	0.5
Скорость поворота башни, град/тик	1	0.75	1.5
Макс. угол поворота башни, град	—	—	20
Макс. здоровье экипажа	100	100	100
Макс. прочность корпуса	200	250	250
Толщина лобовой брони, мм	175	200	250
Толщина бортовой брони, мм	150	175	125
Толщина кормовой брони, мм	100	100	100
Длительность перезарядки, тиков	150	150	125
Бонус к скорости снаряда, точек/тик	0.0	0.41	0.83

Пояснения к некоторым характеристикам:

- Виртуальная длина ствола — расстояние между центром башни (соответствует центру танка) и концом орудия.
- Масса танка и мощность двигателя приведены в справочных целях. Мобильность того или иного танка вы можете определить экспериментально, «на глазок». Можно лишь сказать, что при одинаковой массе и вдвое большей мощности двигателя танк будет поворачиваться вдвое быстрее и вдвое быстрее набирать скорость из положения покоя (при ненулевой скорости на танк также оказывает влияние сопротивление воздуха). Также масса будет влиять на столкновения между различными объектами.

Помимо геометрического положения, у танка есть 3 параметра, отображающих его текущее состояние: прочность корпуса, здоровье экипажа и количество премиумных снарядов. Максимальное здоровье экипажа — всегда 100 единиц, максимальная же прочность корпуса зависит от типа танка. Если значение прочности корпуса или здоровья экипажа опускается до нуля, танк считается

уничтоженным и далее не управляется стратегией. Тем не менее танк не исчезает из мира, а остаётся как пассивный (но не неподвижный) объект. Уничтоженный танк под воздействием внешних сил может быть смещён и в случае, если он подберёт соответствующий бонус, восстановлен. В этом случае управление над ним восстанавливается. Стоит заметить, что с уменьшением здоровья экипажа эффективность управления танком падает: он начинает медленнее двигаться, поворачивать башню, увеличивается также время перезарядки орудия. Если бы при падении здоровья экипажа до нуля танк не считался бы уничтоженным, то эффективность экипажа была бы вдвое меньше максимальной. Премиумные снаряды лучше обычных по большинству параметров, но их количество ограничено, в то время как количество обычных снарядов бесконечно.

## 2.4 Описание снарядов и механики попаданий

Пожалуй, единственным преимуществом обычных снарядов является высокая начальная скорость полёта. Недостатков же можно назвать массу: это и падение бронепробиваемости пропорционально падению скорости, и небольшой урон, и вероятность рикошета.

Премиумные снаряды всех этих недостатков лишены, однако начальная скорость полёта у них меньше, что делает очень проблематичной стрельбу на большие дистанции. Да и количество таких снарядов строго ограничено (подробнее об этом в следующей секции), а в начале игры и вовсе равно нулю.

Основные характеристики снарядов можно увидеть в следующей таблице:

Характеристика снаряда	Обычный снаряд	Премиумный снаряд
Длина, точек	22.5	22.5
Ширина, точек	7.5	7.5
Масса, единиц массы	1	0.5
Начальная скорость полёта, точек/тик	16.67	13.33
Урон	20	35
Бронепробиваемость, мм	215	250
Падение бронепробиваемости	Да	Нет
Угол рикошета, град	60	—

Пояснения к некоторым характеристикам:

- Бронепробиваемость — максимальная толщина брони, которую может пробить снаряд. Помимо типа снаряда на бронепробиваемость влияют и случайные факторы: отклонение реальной бронепробиваемости конкретного снаряда от эталонной бронепробиваемости этого типа снарядов может достигать 20% по модулю. Также следует учесть, что эффективная толщина брони зависит от угла попадания: минимальное (эталонное) значение она принимает только тогда, когда снаряд попадает в танк под углом 0 градусов (угол откладывается от нормали к поверхности танка), и растёт с ростом этого угла, стремясь к бесконечности при угле 90 градусов. Что же зависит от факта пробития? Если снаряд попал в танк и пробил его, урон наносится как корпусу танка, так и экипажу (оба значения равны табличному урону снаряда), иначе — только корпусу.
- Падение бронепробиваемости — свойство снаряда, которое показывает, влияет ли скорость его полёта на бронепробиваемость. Если да, то снаряд только вначале своего полёта имеет максимальную бронепробиваемость, затем она падает пропорционально падению скорости. Стоит заметить, что премиумные снаряды не теряют бронепробиваемости при падении скорости. Также стоит заметить, что бронепробиваемость обычных снарядов имеет табличное значение при табличном же значении скорости. У танков, обладающих бонусом к начальной скорости снаряда, начальная бронепробиваемость выше указанной.
- Угол рикошета — максимальный угол (от нормали к поверхности танка), при котором снаряд считается попавшим в танк. Если угол попадания превышает угол рикошета, то снаряд просто отскакивает от танка и отлетает в сторону. Премиумные снаряды не рикошетят.

## 2.5 Описание бонусов

Во время игры на поле боя случайным образом появляются бонусы — небольшие «коробочки» размером  $30 \times 30$ .

Типы бонусов:

- Аптечка — восстанавливает 35 единиц здоровья экипажа. Здоровье не может превысить отметку 100. Активные танки в процессе боя могут натолкнуть уничтоженный танк на аптечку. В этом случае, если прочность корпуса танка не равна нулю, танк считается восстановленным. Стратегия игрока вновь получает возможность управлять им.
- Ремкомплект — восстанавливает 50 единиц прочности корпуса танка. Прочность корпуса не может превышать максимальной границы, определённой для каждого типа танка. Уничтоженный танк (с разрушенным корпусом) может быть «натолкнут» на ремкомплект. В этом случае, если здоровье экипажа не равно нулю, танк считается восстановленным. Стратегия игрока вновь получает возможность управлять им.
- Ящик с боеприпасами — редкий бонус. Танк получает в свой боекомплект 3 премиумных снаряда, которые он может использовать на своё усмотрение.

Если бонус не был подобран, то он автоматически убирается из игрового мира по прошествии 1000 тиков с момента его появления. Количество бонусов на игру зависит только от её длительности (в тиках), поэтому игроки, сражающиеся в одном бою, не могут получить преимущество (в наборе баллов) перед игроками из другого боя вследствие различного количества бонусов на поле.

## 2.6 Начисление баллов

Вернёмся же к теме начисления баллов игрокам. Надо сказать, что количество набранных игроками баллов — единственный фактор, который определяет победителя. На рейтинг же игрока в рамках турнира влияет только занятое им в бою место.

Список действий, выполняя которые в бою игрок зарабатывает себе баллы:

- Нанесение урона танкам других игроков — за каждую единицу урона корпусу или экипажу танка игрок получает 1 балл.
- Уничтожение танка другого игрока — за каждую уничтоженную боевую машину игрок получает 25 баллов (в дополнение к баллам, полученным за урон).
- Восстановление собственного танка — за восстановление своего танка игрок получает премию в 50 баллов.
- Уничтожение всех танков всех других игроков — за уничтожение всех вражеских танков игрок получает 100 баллов. Следует помнить, что уничтожение всех противников не означает автоматически победу в игре. Победителем считается тот игрок, который набрал максимальное количество баллов.

## Глава 3

# Создание стратегии

### 3.1 Техническая часть

Сперва для создания стратегии вам необходимо выбрать один из ряда поддерживаемых языков программирования: Java (Oracle JDK 6), C# (Mono 2.10+), C++ (GNU MinGW C++ 4.4), Python 2 (Python 2.7+), Python 3 (Python 3.2+), Pascal (Free Pascal 2.6). Возможно, этот набор будет расширен. На сайте проекта вы можете скачать пользовательский пакет для каждого из языков. Модифицировать в пакете разрешено лишь один файл, который и предназначен для содержания вашей стратегии, например, MyStrategy.java (для Java) или MyStrategy.py (для Python). Все остальные файлы пакета при сборке стратегии будут замещены стандартными версиями. Однако вы можете добавлять в стратегию свои файлы с кодом. Эти файлы должны находиться в том же каталоге, что и основной файл стратегии. При отправке решения все они должны быть помещены в один ZIP-архив (файлы должны находиться в корне архива). Если вы не добавляете новых файлов в пакет, достаточно отправить сам файл стратегии (с помощью диалога выбора файла) или же вставить его код в текстовое окно.

После того, как вы отправили свою стратегию, она попадает в очередь тестирования. Система сперва попытается скомпилировать пакет с вашими файлами, а затем, если первая операция прошла успешно, создать несколько коротких (по 100 тиков) игр разных форматов:  $6 \times 1$ ,  $3 \times 2$  и  $2 \times 3$ . Для управления каждой командой танков будет создан отдельный экземпляр вашей стратегии, и для того, чтобы стратегия считалась принятой (корректной) ни один из экземпляров стратегии не должен «упасть». Игрокам в этих тестовых сражениях будут даны имена в формате <имя\_игрока>, <имя\_игрока> (2), <имя\_игрока> (3) и т.д.

После успешного прохождения описанного процесса ваша посылка получает статус «Принята». Первая успешная посылка одновременно означает и вашу регистрацию в Песочнице. Вам начисляется стартовый рейтинг (1200), и ваша стратегия начинает участвовать в периодических квалификационных боях (смотрите описание Песочницы для более подробной информации). Также вам становится доступна функция создания собственных игр, в которых в качестве соперника можно выбирать любую стратегию любого игрока (в том числе и вашу собственную), созданную до момента вашей последней успешной посылки. Созданные вами игры не влияют на рейтинг.

В системе присутствуют ограничения на количество посылок и пользовательских игр, а именно:

- Нельзя отправлять стратегию чаще, чем три раза за пять минут.
- За пять минут нельзя создать более двух пользовательских игр.

Для упрощения отладки небольших изменений стратегии в системе присутствует возможность сделать тестовую посылку (флажок «Тестовая посылка» на форме отправки стратегии). Тестовая посылка не отображается другим пользователям, не участвует в квалификационных боях в Песочнице и боях в этапах турнира, также невозможно собственноручно создавать бои с её участием.

Однако после принятия данной посылки система автоматически добавляет тестовую игру с тремя участниками: непосредственно тестовой посылкой, стратегией, которая ничего не делает, и стратегией из раздела «Быстрый старт». Тестовая игра видна только участнику, сделавшему данную тестовую посылку. Максимальная длительность такой тестовой игры составляет 2000 тиков. На частоту тестовых посылок действует то же ограничение, что и на частоту обычных посылок. Тестовые игры на частоту создания игр пользователем не влияют.

У игроков есть возможность в специальном рендерере просматривать прошедшие бои и даже бои, которые начали тестироваться системой, но результат их ещё не известен (в этом случае данные о ходе боя будут поступать в рендерер по мере обработки их системой). Для этого нужно нажать кнопку «Смотреть» в списке боёв либо нажать кнопку «Посмотреть игру» на странице боя.

Если вы смотрите бой с участием вашей стратегии и заметили некоторую странность в её поведении, или ваша стратегия делает не то, что вы от неё ожидали, то вы можете воспользоваться специальной утилитой Repeater для воспроизведения локального повтора данного боя. Локальный повтор игры — это возможность запустить стратегию на вашем компьютере так, чтобы она видела игровой мир вокруг себя таким, каким он был при тестировании на сервере. Это поможет вам выполнять отладку, добавлять логгирование и наблюдать за реакцией вашей стратегии в каждый момент игры. Для этого скачайте Repeater с сайта CodeTanks 2012 (раздел «Документация» → «Утилита Repeater») и разархивируйте. Для запуска Repeater вам необходимо установленное ПО Java 6 Runtime Environment. Обратите внимание, что любое взаимодействие вашей стратегии с игровым миром при локальном повторе полностью игнорируется. Это означает, что в каждый момент времени окружающий мир для стратегии в точности совпадает с миром, каким он был в игре при тестировании на сервере и не зависит от того, какие действия ваша стратегия предпринимает. Подробнее об утилите Repeater читайте в соответствующем разделе на сайте.

Помимо всего выше перечисленного у игроков есть возможность запускать простые тестовые игры локально на своём компьютере. Для этого необходимо загрузить архив с утилитой Local runner из раздела сайта «Документация» → «Local runner». Использование данной утилиты позволит вам тестировать свою стратегию в условиях, аналогичных условиям тестовой игры на сайте, но без каких-либо ограничений по количеству создаваемых игр. Длительность подобных локальных игр составляет стандартные 5000 тиков. Рендерер для локальных игр заметно отличается от рендерера на сайте. Все игровые объекты в нём отображаются схематично (без использования красочных моделей) в соответствии с их реальными размером и формой, что, возможно, поможет лучше понять взаимодействие этих объектов. Создать локальную тестовую игру очень просто: запустите Local runner с помощью соответствующего скрипта запуска (для Windows или \*n\*x систем), затем запустите свою стратегию из среды разработки (или любым другим удобным вам способом) и смотрите бои. Во время локальных игр вы можете выполнять отладку своей стратегии, ставить точки останова. Однако следует помнить, что Local runner ожидает отклика от стратегии не более 10 минут. По прошествии этого времени он посчитает стратегию «упавшей» и продолжит работу без неё.

## 3.2 Управление танком

Стратегия управляет движением танка, определяя, какую силу приложить к каждой из его гусениц. При этом она не может задать непосредственное значение силы, а лишь указать направление и часть (от  $-1.0$  до  $1.0$ ) мощности двигателя, прикладываемую к гусенице. Задавая для обеих гусениц значение  $1.0$ , мы получаем команду «полный вперёд», задавая  $-1.0$  — «полный назад». Для разворота на месте необходимо приложить к гусеницам силу, одинаковую по модулю, но различную по направлению. Здесь следует учитывать фактор мощности двигателя при движении назад, например, для разворота ТТ (фактор равен  $0.65$ ) вокруг своей оси необходимо задать значения  $0.65$  и  $-1.0$ . Задавая же значения  $-1.0$  и  $1.0$ , мы получаем разворот (более быстрый, чем при  $0.65$  и  $-1.0$ ) одновременно с поступательным движением вперёд.

Помимо управления движением стратегия также может задавать, на какой угол необходимо повернуть орудие/башню в данный тик. Положительное значение — поворот по часовой стрелке,

отрицательное — против. Для каждого типа танка определён модуль максимального угла поворота башни за один тик (значения модулей были приведены ранее в таблице), причём это значение падает с уменьшением здоровья экипажа (как и большинство других характеристик танка). Нужно понимать, что абсолютный поворот башни за тик складывается из двух составляющих: собственного поворота башни (относительно центра корпуса танка) и поворота, вызванного вращением корпуса танка.

Третье и последнее действие, которое может произвести стратегия — выстрел из орудия. Стратегия может выстрелить обычным или премиумным снарядом. Также доступен специальный режим стрельбы, при котором производится выстрел премиумным снарядом, если он есть, а, в случае его отсутствия, — обычным.

Управление танком осуществляется с помощью метода `«move»`<sup>1</sup> стратегии, который вызывается один раз за тик для каждого танка. В качестве параметров методу передаются танк `«Tank self»`, для которого вызывается метод, текущее состояние мира `«World world»` и объект `«Move move»`, устанавливая свойства которого, стратегия и определяет поведение танка. Помимо этого у стратегии есть метод `«selectTank»`, который вызывается один раз для каждого танка до начала отсчёта игрового времени и позволяет определить его тип по номеру `«int tankIndex»`<sup>2</sup> и количеству танков в команде `«int teamSize»`.

---

<sup>1</sup>Здесь и далее используется стиль именования Java, если только явно не указано, о каком языке программирования идёт речь.

<sup>2</sup>Нумерация начинается с нуля.



## 3.3 Примеры реализации

Далее для всех языков программирования приведены простейшие примеры стратегий, которые постоянно поворачивают танк и его башню и стреляют во все стороны. Полную документацию классов и методов для языка Java можно найти в следующих главах.

### 3.3.1 Пример для Java

```
import model.*;

import static java.lang.StrictMath.PI;

public final class MyStrategy implements Strategy {
    @Override
    public void move(Tank self, World world, Move move) {
        move.setLeftTrackPower(-1.0D);
        move.setRightTrackPower(1.0D);
        move.setTurretTurn(PI);
        move.setFireType(FireType.PREMIUM_PREFERRED);
    }

    @Override
    public TankType selectTank(int tankIndex, int teamSize) {
        return TankType.MEDIUM;
    }
}
```

### 3.3.2 Пример для C#

```
using System;
using Com.CodeGame.CodeTanks2012.DevKit.CSharpCgdk.Model;

namespace Com.CodeGame.CodeTanks2012.DevKit.CSharpCgdk
{
    public sealed class MyStrategy : IStrategy
    {
        public void Move(Tank self, World world, Move move)
        {
            move.LeftTrackPower = -1.0D;
            move.RightTrackPower = 1.0D;
            move.TurretTurn = Math.PI;
            move.FireType = FireType.PremiumPreferred;
        }

        public TankType SelectTank(int tankIndex, int teamSize)
        {
            return TankType.Medium;
        }
    }
}
```

### 3.3.3 Пример для C++

```
#include "MyStrategy.h"

#define _USE_MATH_DEFINES
#include <cmath>

using namespace model;

void MyStrategy::Move(Tank self, World world, model::Move& move) {
    move.set_left_track_power(-1.0);
    move.set_right_track_power(1.0);
    move.set_turret_turn(M_PI);
    move.set_fire_type(PREMIUM_PREFERRED);
}

TankType MyStrategy::SelectTank(int tank_index, int team_size) {
    return MEDIUM;
}
```

### 3.3.4 Пример для Python 2 и Python 3

В языке Python имя переменной текущего танка изменено с «self» на «me».

```
from math import *
from model.FireType import FireType
from model.TankType import TankType

class MyStrategy:
    def move(self, me, world, move):
        move.left_track_power = -1.0
        move.right_track_power = 1.0
        move.turret_turn = pi
        move.fire_type = FireType.PREMIUM_PREFERRED

    def select_tank(self, tank_index, team_size):
        return TankType.MEDIUM
```

### 3.3.5 Пример для Pascal

В языке Pascal имя переменной текущего танка изменено с «self» на «me».

```
unit MyStrategy;

interface

uses
    StrategyControl, TankControl, WorldControl, MoveControl;

type
    TMyStrategy = class(TStrategy)
    public
```

```

        procedure Move(me: TTank; world: TWorld; var move: TMove); override;
        function SelectTank(tankIndex: longint; teamSize: longint) : TTankType; override;

    end;

implementation

uses
    Math;

procedure TMyStrategy.Move(me: TTank; world: TWorld; var move: TMove);
begin
    move.SetLeftTrackPower(-1.0);
    move.SetRightTrackPower(1.0);
    move.SetTurretTurn(-PI);
    move.SetFireType(PREMIUM_PREFERRED);
end;

function TMyStrategy.SelectTank(tankIndex: longint; teamSize: longint) : TTankType;
begin
    Result := HEAVY;
end;

end.

```

## Глава 4

# Package model

Package Contents

Page

---

### Classes

<b>Bonus</b> .....	19
<i>Класс, определяющий бонус — неподвижный полезный объект.</i>	
<b>BonusType</b> .....	19
<i>Тип бонуса.</i>	
<b>FireType</b> .....	20
<i>Тип выстрела.</i>	
<b>Move</b> .....	21
<i>Стратегия игрока может управлять танком посредством установки свойств объекта данного класса.</i>	
<b>Obstacle</b> .....	22
<i>Класс, определяющий препятствие — неподвижный неразрушимый объект.</i>	
<b>Player</b> .....	22
<i>Содержит данные о текущем состоянии игрока.</i>	
<b>Shell</b> .....	23
<i>Класс, определяющий снаряд.</i>	
<b>ShellType</b> .....	23
<i>Тип снаряда.</i>	
<b>Tank</b> .....	24
<i>Класс, определяющий танк.</i>	
<b>TankType</b> .....	27
<i>Тип танка.</i>	
<b>Unit</b> .....	27
<i>Базовый класс для определения объектов («юнитов») на игровом поле.</i>	
<b>World</b> .....	29
<i>Содержит описание игрового мира и позволяет получить списки различных юнитов, присутствующих на поле боя.</i>	

---

## 4.1 Classes

### 4.1.1 CLASS Bonus

---

Класс, определяющий бонус — неподвижный полезный объект. Содержит также все свойства юнита.

#### DECLARATION

---

```
public final class Bonus
extends Unit
```

#### METHODS

---

- *getType*  
`public BonusType getType( )`  
– **Returns** - Возвращает тип бонуса.

### 4.1.2 CLASS BonusType

---

Тип бонуса.

#### DECLARATION

---

```
public final class BonusType
extends Enum
```

#### FIELDS

---

- `public static final BonusType MEDIKIT`  
– Аптечка, восстанавливает здоровье экипажа.
- `public static final BonusType REPAIR_KIT`  
– Ремкомплект, восстанавливает прочность корпуса танка.
- `public static final BonusType AMMO_CRATE`  
– Ящик с боеприпасами, содержит несколько премиумных снарядов.

## METHODS

---

- *valueOf*  
public static BonusType valueOf( String name )
- *values*  
public static BonusType[] values( )

### 4.1.3 CLASS FireType

---

Тип выстрела.

## DECLARATION

---

```
public final class FireType
extends Enum
```

## FIELDS

---

- public static final FireType NONE  
– Не стрелять.
- public static final FireType REGULAR  
– Выстрелить обычным снарядам.
- public static final FireType PREMIUM  
– Выстрелить премиумным снарядам. При его отсутствии не делать ничего.
- public static final FireType PREMIUM\_PREFERRED  
– Выстрелить премиумным снарядам. При его отсутствии выстрелить обычным снарядам.

## METHODS

---

- *valueOf*  
public static FireType valueOf( String name )
- *values*  
public static FireType[] values( )

#### 4.1.4 CLASS Move

---

Стратегия игрока может управлять танком посредством установки свойств объекта данного класса.

##### DECLARATION

---

```
public final class Move  
  
extends Object
```

##### METHODS

---

- *getFireType*  
public FireType **getFireType**( )  
    – **Returns** - Возвращает текущий режим стрельбы.
- *getLeftTrackPower*  
public double **getLeftTrackPower**( )  
    – **Returns** - Возвращает текущую установку мощности двигателя для левой гусеницы.
- *getRightTrackPower*  
public double **getRightTrackPower**( )  
    – **Returns** - Возвращает текущую установку мощности двигателя для правой гусеницы.
- *getTurretTurn*  
public double **getTurretTurn**( )  
    – **Returns** - Возвращает текущую установку для поворота башни.
- *setFireType*  
public void **setFireType**( FireType fireType )  
    – **Usage**  
        \* Устанавливает режим стрельбы.  
    – **Parameters**  
        \* fireType - Режим стрельбы.
- *setLeftTrackPower*  
public void **setLeftTrackPower**( double leftTrackPower )  
    – **Usage**  
        \* Задаёт мощность двигателя для левой гусеницы.  
    – **Parameters**  
        \* leftTrackPower - Значение в интервале [-1.0, 1.0], где -1.0 означает «полный назад», а 1.0 — «полный вперёд». Значения, лежащие за пределами интервала, будут приведены к ближайшей его границе.

- *setRightTrackPower*

```
public void setRightTrackPower( double rightTrackPower )
```

- **Usage**

- \* Задаёт мощность двигателя для правой гусеницы.

- **Parameters**

- \* **rightTrackPower** - Значение в интервале  $[-1.0, 1.0]$ , где  $-1.0$  означает «полный назад», а  $1.0$  — «полный вперёд». Значения, лежащие за пределами интервала, будут приведены к ближайшей его границе.

---

- *setTurretTurn*

```
public void setTurretTurn( double turretTurn )
```

- **Usage**

- \* Задаёт направление и скорость поворота башни в следующий тик.

- **Parameters**

- \* **turretTurn** - Значение в интервале  $[-a, a]$ , где  $a$  — табличное значение скорости поворота башни для данного типа танка. Положительные значения соответствуют повороту по часовой стрелке, отрицательные — против. Значения, лежащие за пределами интервала, будут приведены к ближайшей его границе.

#### 4.1.5 CLASS Obstacle

---

Класс, определяющий препятствие — неподвижный неразрушимый объект.

DECLARATION

---

```
public final class Obstacle
extends Unit
```

#### 4.1.6 CLASS Player

---

Содержит данные о текущем состоянии игрока.

DECLARATION

---

```
public final class Player
extends Object
```



## METHODS

---

- *getName*  
`public String getName( )`  
— **Returns** - Возвращает имя игрока.
- *getScore*  
`public int getScore( )`  
— **Returns** - Возвращает текущее количество баллов, набранных игроком.
- *isStrategyCrashed*  
`public boolean isStrategyCrashed( )`  
— **Returns** - Возвращает специальный флаг — показатель того, что стратегия игрока «упала». Более подробную информацию можно найти в документации к игре.

### 4.1.7 CLASS Shell

---

Класс, определяющий снаряд. Содержит также все свойства юнита.

## DECLARATION

---

```
public final class Shell
extends Unit
```

## METHODS

---

- *getPlayerName*  
`public String getPlayerName( )`  
— **Returns** - Возвращает имя игрока — владельца танка, выпустившего снаряд.
- *getType*  
`public ShellType getType( )`  
— **Returns** - Возвращает тип снаряда.

### 4.1.8 CLASS ShellType

---

Тип снаряда.

## DECLARATION

---

```
public final class ShellType  
extends Enum
```

## FIELDS

- `public static final ShellType REGULAR`
  - Обычный.
- `public static final ShellType PREMIUM`
  - Премиумный.

## METHODS

- *valueOf*  
`public static ShellType valueOf( String name )`
- *values*  
`public static ShellType[] values( )`

### 4.1.9 CLASS Tank

---

Класс, определяющий танк. Содержит также все свойства юнита.

## DECLARATION

---

```
public final class Tank  
extends Unit
```

## METHODS

- *getCrewHealth*  
`public int getCrewHealth( )`
    - **Returns** - Возвращает текущее здоровье экипажа.
-

- *getCrewMaxHealth*  
 public int **getCrewMaxHealth**( )  


---

 — **Returns** - Возвращает максимальное здоровье экипажа.
- *getEnginePower*  
 public double **getEnginePower**( )  


---

 — **Returns** - Возвращает мощность двигателя в единицах мощности.
- *getEngineRearPowerFactor*  
 public double **getEngineRearPowerFactor**( )  


---

 — **Returns** - Возвращает коэффициент мощности двигателя при движении назад.
- *getFrontalArmor*  
 public int **getFrontalArmor**( )  


---

 — **Returns** - Возвращает толщину лобовой брони танка в мм.
- *getHullDurability*  
 public int **getHullDurability**( )  


---

 — **Returns** - Возвращает текущую прочность корпуса.
- *getHullMaxDurability*  
 public int **getHullMaxDurability**( )  


---

 — **Returns** - Возвращает максимальную прочность корпуса.
- *getMass*  
 public double **getMass**( )  


---

 — **Returns** - Возвращает массу танка в единицах массы.
- *getPlayerName*  
 public String **getPlayerName**( )  


---

 — **Returns** - Возвращает имя игрока — владельца танка.
- *getPremiumShellCount*  
 public int **getPremiumShellCount**( )  


---

 — **Returns** - Возвращает количество премиумных снарядов.
- *getRearArmor*  
 public int **getRearArmor**( )  


---

 — **Returns** - Возвращает толщину кормовой брони танка в мм.
- *getReloadingTime*  
 public int **getReloadingTime**( )  


---

 — **Returns** - Возвращает количество времени в тиках, необходимое для выполнения полной перезарядки орудия.
- *getRemainingReloadingTime*  
 public int **getRemainingReloadingTime**( )  


---

 — **Returns** - Возвращает количество времени в тиках, оставшееся до следующего возможного выстрела. Если значение равно нулю, танк может произвести выстрел в данный тик.

- *getShellSpeedBonus*  
`public double getShellSpeedBonus( )`  


---

  - **Returns** - Возвращает бонус к начальной скорости снаряда в точках за тик.
- *getSideArmor*  
`public int getSideArmor( )`  


---

  - **Returns** - Возвращает толщину бортовой брони танка в мм.
- *getTeammateIndex*  
`public int getTeammateIndex( )`  


---

  - **Returns** - Возвращает 0-индексированный номер танка в команде.
- *getTurretAngleTo*  
`public double getTurretAngleTo( double x, double y )`  


---

  - **Parameters**
    - \* x - X-координата точки.
    - \* y - Y-координата точки.
  - **Returns** - Возвращает ориентированный угол  $[-\pi, \pi]$  между орудием танка и вектором из центра танка (начала орудия) к указанной точке.
- *getTurretAngleTo*  
`public double getTurretAngleTo( Unit unit )`  


---

  - **Parameters**
    - \* unit - Объект, к центру которого необходимо определить угол.
  - **Returns** - Возвращает ориентированный угол  $[-\pi, \pi]$  между орудием танка и вектором из центра танка (начала орудия) к центру указанного объекта.
- *getTurretMaxRelativeAngle*  
`public double getTurretMaxRelativeAngle( )`  


---

  - **Returns** - Возвращает максимально возможный угол (в радианах) поворота башни относительно танка или 0.0D, если танк не имеет данного ограничения.
- *getTurretRelativeAngle*  
`public double getTurretRelativeAngle( )`  


---

  - **Returns** - Возвращает угол (в радианах) поворота башни относительно танка.
- *getTurretTurnSpeed*  
`public double getTurretTurnSpeed( )`  


---

  - **Returns** - Возвращает скорость поворота башни в радианах за тик.
- *getType*  
`public TankType getType( )`  


---

  - **Returns** - Возвращает тип танка.
- *getVirtualGunLength*  
`public double getVirtualGunLength( )`  


---

  - **Returns** - Возвращает виртуальную длину орудия — расстояние от центра танка до центра снаряда в момент выстрела.
- *isTeammate*  
`public boolean isTeammate( )`  


---

  - **Returns** - Возвращает `true` если и только если данный танк является дружественным.

#### 4.1.10 CLASS TankType

---

Тип танка.

##### DECLARATION

---

```
public final class TankType
extends Enum
```

##### FIELDS

---

- `public static final TankType MEDIUM`  
– Средний танк, СТ.
- `public static final TankType HEAVY`  
– Тяжёлый танк, ТТ.
- `public static final TankType TANK_DESTROYER`  
– Самоходное противотанковое орудие, ПТ.

##### METHODS

---

- *valueOf*  
`public static TankType valueOf( String name )`
- *values*  
`public static TankType[] values( )`

#### 4.1.11 CLASS Unit

---

Базовый класс для определения объектов («юнитов») на игровом поле.

##### DECLARATION

---

```
public abstract class Unit
extends Object
```

- *getAngle*

```
public final double getAngle( )
```

- **Returns** - Возвращает угол поворота объекта в радианах. Нулевой угол соответствует направлению оси абсцисс. Положительные значения соответствуют повороту по часовой стрелке.
- 

- *getAngleTo*

```
public double getAngleTo( double x, double y )
```

- **Parameters**

- \* x - X-координата точки.
- \* y - Y-координата точки.

- **Returns** - Возвращает ориентированный угол  $[-\pi, \pi]$  между направлением данного объекта и вектором из центра данного объекта к указанной точке.
- 

- *getAngleTo*

```
public double getAngleTo( Unit unit )
```

- **Parameters**

- \* unit - Объект, к центру которого необходимо определить угол.

- **Returns** - Возвращает ориентированный угол  $[-\pi, \pi]$  между направлением данного объекта и вектором из центра данного объекта к центру указанного объекта.
- 

- *getAngularSpeed*

```
public double getAngularSpeed( )
```

- **Returns** - Возвращает скорость вращения объекта. Положительные значения соответствуют вращению по часовой стрелке.
- 

- *getDistanceTo*

```
public double getDistanceTo( double x, double y )
```

- **Parameters**

- \* x - X-координата точки.
- \* y - Y-координата точки.

- **Returns** - Возвращает расстояние до точки от центра данного объекта.
- 

- *getDistanceTo*

```
public double getDistanceTo( Unit unit )
```

- **Parameters**

- \* unit - Объект, до центра которого необходимо определить расстояние.

- **Returns** - Возвращает расстояние от центра данного объекта до центра указанного объекта.
- 

- *getHeight*

```
public double getHeight( )
```

- **Returns** - Возвращает высоту объекта.
- 

- *getId*

```
public long getId( )
```

- **Returns** - Возвращает уникальный идентификатор объекта.
-

- *getSpeedX*  
`public final double getSpeedX( )`  
 – **Returns** - Возвращает X-составляющую скорости объекта. Ось абсцисс направлена слева направо.

---

- *getSpeedY*  
`public final double getSpeedY( )`  
 – **Returns** - Возвращает Y-составляющую скорости объекта. Ось ординат направлена сверху вниз.

---

- *getWidth*  
`public double getWidth( )`  
 – **Returns** - Возвращает ширину объекта.

---

- *getX*  
`public final double getX( )`  
 – **Returns** - Возвращает X-координату центра объекта. Ось абсцисс направлена слева направо.

---

- *getY*  
`public final double getY( )`  
 – **Returns** - Возвращает Y-координату центра объекта. Ось ординат направлена сверху вниз.

#### 4.1.12 CLASS World

---

Содержит описание игрового мира и позволяет получить списки различных юнитов, присутствующих на поле боя.

##### DECLARATION

---

```
public final class World
extends Object
```

##### METHODS

---

- *getBonuses*  
`public Bonus[] getBonuses( )`  
 – **Returns** - Возвращает список бонусов (в случайном порядке). После каждого тика объекты, задающие бонусы, пересоздаются.

---

- *getHeight*  
`public double getHeight( )`  
 – **Returns** - Возвращает высоту мира в точках.

- 
- *getObstacles*  
`public Obstacle[] getObstacles( )`
    - **Returns** - Возвращает список препятствий (в случайном порядке). После каждого тика объекты, задающие препятствия, пересоздаются.
  - *getPlayers*  
`public Player[] getPlayers( )`
    - **Returns** - Возвращает список игроков (в случайном порядке). После каждого тика объекты, задающие игроков, пересоздаются.
  - *getShells*  
`public Shell[] getShells( )`
    - **Returns** - Возвращает список снарядов (в случайном порядке). После каждого тика объекты, задающие снаряды, пересоздаются.
  - *getTanks*  
`public Tank[] getTanks( )`
    - **Returns** - Возвращает список танков (в случайном порядке), включая танк стратегии, вызвавшей этот метод. После каждого тика объекты, задающие танки, пересоздаются.
  - *getTick*  
`public int getTick( )`
    - **Returns** - Возвращает номер текущего тика.
  - *getWidth*  
`public double getWidth( )`
    - **Returns** - Возвращает ширину мира в точках.



## Глава 5

### Package <none>

*Package Contents*

*Page*

---

#### Interfaces

Strategy ..... 32

*Стратегия — интерфейс, содержащий описание методов искусственного ин-  
теллекта танка.*

---

## 5.1 Interfaces

### 5.1.1 INTERFACE Strategy

---

Стратегия — интерфейс, содержащий описание методов искусственного интеллекта танка. Каждая пользовательская стратегия должна реализовывать этот интерфейс. Может отсутствовать в некоторых языковых пакетах, если язык не поддерживает интерфейсы.

#### DECLARATION

---

```
public interface Strategy
```

#### METHODS

---

- *move*

```
public void move( Tank self, World world, Move move )
```

- **Usage**

- \* Основной метод стратегии, осуществляющий управление танком. Вызывается каждый тик для каждого танка.

- **Parameters**

- \* **self** - Танк, которым данный метод будет осуществлять управление.
    - \* **world** - Текущее состояние мира.
    - \* **move** - Результатом работы метода является изменение полей данного объекта.

---

- *selectTank*

```
public TankType selectTank( int tankIndex, int teamSize )
```

- **Usage**

- \* Позволяет выбрать тип танка. Вызывается один раз для каждого танка до начала отсчёта игрового времени.

- **Parameters**

- \* **tankIndex** - 0-индексированный номер танка в команде.
    - \* **teamSize** - Размер команды.

- **Returns** - Возвращает выбранный тип танка.