**4.1.5** CLASS **Game**

Предоставляет доступ к различным игровым константам.

DECLARATION

public class Game **extends** Object

METHODS

* *getActionDetectionInterval* public int **getActionDetectionInterval**( )
  + **Returns** - Возвращает интервал, учитываемый в ограничении количества действий стратегии.
* *getAdditionalActionCountPerControlCenter* public int **getAdditionalActionCountPerControlCenter**( )
  + **Returns** - Возвращает дополнительное количество действий за каждый захваченный центр управления (FacilityType.CONTROL\_CENTER).
* *getArrvAerialDefence* public int **getArrvAerialDefence**( )

**– Returns** - Возвращает защиту БРЭМ от атак воздушной техники.

* *getArrvDurability* public int **getArrvDurability**( )
  + **Returns** - Возвращает максимальную прочность БРЭМ.
* *getArrvGroundDefence* public int **getArrvGroundDefence**( )
  + **Returns** - Возвращает защиту БРЭМ от атак наземной техники.
* *getArrvProductionCost* public int **getArrvProductionCost**( )
  + **Returns** - Возвращает количество тиков, необхожимое для производства одной БРЭМ на заводе (FacilityType.VEHICLE\_FACTORY).
* *getArrvRepairRange* public double **getArrvRepairRange**( )
  + **Returns** - Возвращает максимальное расстояние (от центра до центра), на котором БРЭМ может ремонтировать дружественную технику.
* *getArrvRepairSpeed* public double **getArrvRepairSpeed**( )
  + **Returns** - Возвращает максимальное количество прочности, которое БРЭМ может восстановить дружественной технике за один тик.
* *getArrvSpeed* public double **getArrvSpeed**( )
  + **Returns** - Возвращает максимальную скорость БРЭМ.
* *getArrvVisionRange* public double **getArrvVisionRange**( )
  + **Returns** - Возвращает базовый радиус обзора БРЭМ.
* *getBaseActionCount* public int **getBaseActionCount**( )
  + **Returns** - Возвращает базовое количество действий, которое может совершить стратегия за actionDetectionInterval последовательных тиков.
* *getBaseTacticalNuclearStrikeCooldown* public int **getBaseTacticalNuclearStrikeCooldown**( )
  + **Returns** - Возвращает минимально возможный интервал между двумя последовательными тактическими ядерными ударами (ActionType.TACTICAL\_NUCLEAR\_STRIKE).
* *getClearWeatherSpeedFactor* public double **getClearWeatherSpeedFactor**( )
  + **Returns** - Возвращает мультипликатор максимальной скорости воздушной техники, находящейся в области ясной погоды (WeatherType.CLEAR).
* *getClearWeatherStealthFactor* public double **getClearWeatherStealthFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора любой техники при обнаружении воздушной техники противника, находящейся в области ясной погоды (WeatherType.CLEAR).
* *getClearWeatherVisionFactor* public double **getClearWeatherVisionFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора воздушной техники, находящейся в области ясной погоды (WeatherType.CLEAR).
* *getCloudWeatherSpeedFactor* public double **getCloudWeatherSpeedFactor**( )
  + **Returns** - Возвращает мультипликатор максимальной скорости воздушной техники, находящейся в плотных облаках (WeatherType.CLOUD).
* *getCloudWeatherStealthFactor* public double **getCloudWeatherStealthFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора любой техники при обнаружении воздушной техники противника, находящейся в плотных облаках (WeatherType.CLOUD).
* *getCloudWeatherVisionFactor* public double **getCloudWeatherVisionFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора воздушной техники, находящейся в плотных облаках (WeatherType.CLOUD).
* *getFacilityCapturePointsPerVehiclePerTick* public double **getFacilityCapturePointsPerVehiclePerTick**( )
  + **Returns** - Возвращает скорость изменения индикатора захвата сооружения

(facility.capturePoints) за каждую единицу техники, центр которой находится внутри сооружения.

* *getFacilityCaptureScore* public int **getFacilityCaptureScore**( )
  + **Returns** - Возвращает количество баллов за захват сооружения.
* *getFacilityHeight* public double **getFacilityHeight**( )
  + **Returns** - Возвращает высоту сооружения.
* *getFacilityWidth* public double **getFacilityWidth**( )
  + **Returns** - Возвращает ширину сооружения.
* *getFighterAerialAttackRange* public double **getFighterAerialAttackRange**( )
  + **Returns** - Возвращает дальность атаки истребителя по воздушным целям.
* *getFighterAerialDamage* public int **getFighterAerialDamage**( )
  + **Returns** - Возвращает урон одной атаки истребителя по воздушной технике.
* *getFighterAerialDefence* public int **getFighterAerialDefence**( )
  + **Returns** - Возвращает защиту истребителя от атак воздушной техники.
* *getFighterAttackCooldownTicks* public int **getFighterAttackCooldownTicks**( )
  + **Returns** - Возвращает интервал в тиках между двумя последовательными атаками истребителя.
* *getFighterDurability* public int **getFighterDurability**( )
  + **Returns** - Возвращает максимальную прочность истребителя.
* *getFighterGroundAttackRange* public double **getFighterGroundAttackRange**( )
  + **Returns** - Возвращает дальность атаки истребителя по наземным целям.
* *getFighterGroundDamage* public int **getFighterGroundDamage**( )
  + **Returns** - Возвращает урон одной атаки истребителя по наземной технике.
* *getFighterGroundDefence* public int **getFighterGroundDefence**( )
  + **Returns** - Возвращает защиту истребителя от атак наземной техники.
* *getFighterProductionCost* public int **getFighterProductionCost**( )
  + **Returns** - Возвращает количество тиков, необхожимое для производства одного истребителя на заводе (FacilityType.VEHICLE\_FACTORY).
* *getFighterSpeed* public double **getFighterSpeed**( )
  + **Returns** - Возвращает максимальную скорость истребителя.
* *getFighterVisionRange* public double **getFighterVisionRange**( )
  + **Returns** - Возвращает базовый радиус обзора истребителя.
* *getForestTerrainSpeedFactor* public double **getForestTerrainSpeedFactor**( )
  + **Returns** - Возвращает мультипликатор максимальной скорости наземной техники, находящейся в лесистой местности (TerrainType.FOREST).
* *getForestTerrainStealthFactor* public double **getForestTerrainStealthFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора любой техники при обнаружении наземной техники противника, находящейся в лесистой местности (TerrainType.FOREST).
* *getForestTerrainVisionFactor* public double **getForestTerrainVisionFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора наземной техники, находящейся в лесистой местности (TerrainType.FOREST).
* *getHelicopterAerialAttackRange* public double **getHelicopterAerialAttackRange**( )
  + **Returns** - Возвращает дальность атаки ударного вертолёта по воздушным целям.
* *getHelicopterAerialDamage* public int **getHelicopterAerialDamage**( )
  + **Returns** - Возвращает урон одной атаки ударного вертолёта по воздушной технике.
* *getHelicopterAerialDefence* public int **getHelicopterAerialDefence**( )
  + **Returns** - Возвращает защиту ударного вертолёта от атак воздушной техники.
* *getHelicopterAttackCooldownTicks* public int **getHelicopterAttackCooldownTicks**( )
  + **Returns** - Возвращает интервал в тиках между двумя последовательными атаками ударного вертолёта.
* *getHelicopterDurability* public int **getHelicopterDurability**( )
  + **Returns** - Возвращает максимальную прочность ударного вертолёта.
* *getHelicopterGroundAttackRange* public double **getHelicopterGroundAttackRange**( )
  + **Returns** - Возвращает дальность атаки ударного вертолёта по наземным целям.
* *getHelicopterGroundDamage* public int **getHelicopterGroundDamage**( )
  + **Returns** - Возвращает урон одной атаки ударного вертолёта по наземной технике.
* *getHelicopterGroundDefence* public int **getHelicopterGroundDefence**( )
  + **Returns** - Возвращает защиту ударного вертолёта от атак наземной техники.
* *getHelicopterProductionCost* public int **getHelicopterProductionCost**( )
  + **Returns** - Возвращает количество тиков, необхожимое для производства одного ударного вертолёта на заводе (FacilityType.VEHICLE\_FACTORY).
* *getHelicopterSpeed* public double **getHelicopterSpeed**( )
  + **Returns** - Возвращает максимальную скорость ударного вертолёта.
* *getHelicopterVisionRange* public double **getHelicopterVisionRange**( )
  + **Returns** - Возвращает базовый радиус обзора ударного вертолёта.
* *getIfvAerialAttackRange* public double **getIfvAerialAttackRange**( )
  + **Returns** - Возвращает дальность атаки БМП по воздушным целям.
* *getIfvAerialDamage* public int **getIfvAerialDamage**( )
  + **Returns** - Возвращает урон одной атаки БМП по воздушной технике.
* *getIfvAerialDefence* public int **getIfvAerialDefence**( )
  + **Returns** - Возвращает защиту БМП от атак воздушной техники.
* *getIfvAttackCooldownTicks* public int **getIfvAttackCooldownTicks**( )
  + **Returns** - Возвращает интервал в тиках между двумя последовательными атаками БМП.
* *getIfvDurability* public int **getIfvDurability**( )
  + **Returns** - Возвращает максимальную прочность БМП.
* *getIfvGroundAttackRange* public double **getIfvGroundAttackRange**( )
  + **Returns** - Возвращает дальность атаки БМП по наземным целям.
* *getIfvGroundDamage* public int **getIfvGroundDamage**( )
  + **Returns** - Возвращает урон одной атаки БМП по наземной технике.
* *getIfvGroundDefence* public int **getIfvGroundDefence**( )
  + **Returns** - Возвращает защиту БМП от атак наземной техники.
* *getIfvProductionCost* public int **getIfvProductionCost**( )
  + **Returns** - Возвращает количество тиков, необхожимое для производства одной БМП на заводе (FacilityType.VEHICLE\_FACTORY).
* *getIfvSpeed* public double **getIfvSpeed**( )
  + **Returns** - Возвращает максимальную скорость БМП.
* *getIfvVisionRange* public double **getIfvVisionRange**( )
  + **Returns** - Возвращает базовый радиус обзора БМП.
* *getMaxFacilityCapturePoints* public double **getMaxFacilityCapturePoints**( )
  + **Returns** - Возвращает максимально возможную абсолютную величину индикатора захвата сооружения (facility.capturePoints).
* *getMaxTacticalNuclearStrikeDamage* public double **getMaxTacticalNuclearStrikeDamage**( )
  + **Returns** - Возвращает урон тактического ядерного удара в центре взрыва.
* *getMaxUnitGroup* public int **getMaxUnitGroup**( )
  + **Returns** - Возвращает максимально возможный индекс группы юнитов.
* *getPlainTerrainSpeedFactor* public double **getPlainTerrainSpeedFactor**( )
  + **Returns** - Возвращает мультипликатор максимальной скорости наземной техники, находящейся на равнинной местности (TerrainType.PLAIN).

*getPlainTerrainStealthFactor* public double **getPlainTerrainStealthFactor**( )

* + **Returns** - Возвращает мультипликатор радиуса обзора любой техники при обнаружении наземной техники противника, находящейся на равнинной местности (TerrainType.PLAIN).
* *getPlainTerrainVisionFactor* public double **getPlainTerrainVisionFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора наземной техники, находящейся на равнинной местности (TerrainType.PLAIN).
* *getRainWeatherSpeedFactor* public double **getRainWeatherSpeedFactor**( )
  + **Returns** - Возвращает мультипликатор максимальной скорости воздушной техники, находящейся в условиях сильного дождя (WeatherType.RAIN).
* *getRainWeatherStealthFactor* public double **getRainWeatherStealthFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора любой техники при обнаружении воздушной техники противника, находящейся в условиях сильного дождя (WeatherType.RAIN).
* *getRainWeatherVisionFactor* public double **getRainWeatherVisionFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора воздушной техники, находящейся в условиях сильного дождя (WeatherType.RAIN).
* *getRandomSeed* public long **getRandomSeed**( )
  + **Returns** - Возвращает некоторое число, которое ваша стратегия может использовать для инициализации генератора случайных чисел. Данное значение имеет рекомендательный характер, однако позволит более точно воспроизводить прошедшие игры.
* *getSwampTerrainSpeedFactor* public double **getSwampTerrainSpeedFactor**( )
  + **Returns** - Возвращает мультипликатор максимальной скорости наземной техники, находящейся в болотистой местности (TerrainType.SWAMP).
* *getSwampTerrainStealthFactor* public double **getSwampTerrainStealthFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора любой техники при обнаружении наземной техники противника, находящейся в болотистой местности (TerrainType.SWAMP).
* *getSwampTerrainVisionFactor* public double **getSwampTerrainVisionFactor**( )
  + **Returns** - Возвращает мультипликатор радиуса обзора наземной техники, находящейся в болотистой местности (TerrainType.SWAMP).
* *getTacticalNuclearStrikeCooldownDecreasePerControlCenter* public int **getTacticalNuclearStrikeCooldownDecreasePerControlCenter**( )
  + **Returns** - Возвращает уменьшение интервала между тактическими ядерными ударами за каждый захваченный центр управления (FacilityType.CONTROL\_CENTER).
* *getTacticalNuclearStrikeDelay* public int **getTacticalNuclearStrikeDelay**( )
  + **Returns** - Возвращает задержку между запросом нанесения тактического ядерного удара и собственно самим нанесением.
* *getTacticalNuclearStrikeRadius* public double **getTacticalNuclearStrikeRadius**( )
  + **Returns** - Возвращает радиус взрыва тактического ядерного удара.
* *getTankAerialAttackRange* public double **getTankAerialAttackRange**( )
  + **Returns** - Возвращает дальность атаки танка по воздушным целям.
* *getTankAerialDamage* public int **getTankAerialDamage**( )
  + **Returns** - Возвращает урон одной атаки танка по воздушной технике.
* *getTankAerialDefence* public int **getTankAerialDefence**( )
  + **Returns** - Возвращает защиту танка от атак воздушной техники.
* *getTankAttackCooldownTicks* public int **getTankAttackCooldownTicks**( )
  + **Returns** - Возвращает интервал в тиках между двумя последовательными атаками танка.
* *getTankDurability* public int **getTankDurability**( )
  + **Returns** - Возвращает максимальную прочность танка.
* *getTankGroundAttackRange* public double **getTankGroundAttackRange**( )
  + **Returns** - Возвращает дальность атаки танка по наземным целям.
* *getTankGroundDamage* public int **getTankGroundDamage**( )
  + **Returns** - Возвращает урон одной атаки танка по наземной технике.
* *getTankGroundDefence* public int **getTankGroundDefence**( )
  + **Returns** - Возвращает защиту танка от атак наземной техники.
* *getTankProductionCost* public int **getTankProductionCost**( )
  + **Returns** - Возвращает количество тиков, необхожимое для производства одного танка на заводе (FacilityType.VEHICLE\_FACTORY).
* *getTankSpeed* public double **getTankSpeed**( )
  + **Returns** - Возвращает максимальную скорость танка.
* *getTankVisionRange* public double **getTankVisionRange**( )
  + **Returns** - Возвращает базовый радиус обзора танка.

*getTerrainWeatherMapColumnCount* public int **getTerrainWeatherMapColumnCount**( )

* + **Returns** - Возвращает количество столбцов в картах местности и погоды.
* *getTerrainWeatherMapRowCount* public int **getTerrainWeatherMapRowCount**( )

**– Returns** - Возвращает количество строк в картах местности и погоды.

* *getTickCount* public int **getTickCount**( )

**– Returns** - Возвращает базовую длительность игры в тиках. Реальная длительность может отличаться от этого значения в меньшую сторону. Эквивалентно world.tickCount.

* *getVehicleEliminationScore* public int **getVehicleEliminationScore**( )

**– Returns** - Возвращает количество баллов за уничтожение юнита противника.

* *getVehicleRadius* public double **getVehicleRadius**( )

**– Returns** - Возвращает радиус техники.

* *getVictoryScore* public int **getVictoryScore**( )

**– Returns** - Возвращает количество баллов, получаемое игроком в случае уничтожения всех юнитов противника.

* *getWorldHeight* public double **getWorldHeight**( )

**– Returns** - Возвращает высоту карты.

* *getWorldWidth* public double **getWorldWidth**( )

**– Returns** - Возвращает ширину карты.

* *isFogOfWarEnabled* public boolean **isFogOfWarEnabled**( )

**– Returns** - Возвращает true, если и только если в данной игре включен режим частичной видимости.