

# Table Of Content

---

<a href="#">IStats</a> .....	2
<a href="#">pl.projekt.game.item</a> .....	2
<a href="#">AbstractItem</a> .....	3
<a href="#">Armor</a> .....	5
<a href="#">ArmorTest</a> .....	6
<a href="#">Axe</a> .....	7
<a href="#">Dagger</a> .....	8
<a href="#">Hammer</a> .....	10
<a href="#">Jewelery</a> .....	11
<a href="#">Maze</a> .....	13
<a href="#">Shield</a> .....	14
<a href="#">Sword</a> .....	15
<a href="#">pl.projekt.game.material</a> .....	17
<a href="#">AbstractMaterials</a> .....	17
<a href="#">Diamond</a> .....	18
<a href="#">Iron</a> .....	19
<a href="#">Stone</a> .....	20
<a href="#">Wood</a> .....	21
<a href="#">pl.projekt.game.mob</a> .....	23
<a href="#">AbstractMonster</a> .....	23
<a href="#">Dwarf</a> .....	26
<a href="#">Elf</a> .....	28
<a href="#">Minotaur</a> .....	30
<a href="#">MinotaurTest</a> .....	33
<a href="#">Orc</a> .....	33
<a href="#">pl.projekt.simulation</a> .....	36
<a href="#">Board</a> .....	36
<a href="#">IRandom</a> .....	38
<a href="#">SimulationApp</a> .....	39
<a href="#">Index</a> .....	41

---

# Interface IStats

---

< [Methods](#) >

---

public interface **IStats**

Interfejs po którym dziedziczÄ... moby oraz przedmioty.

## Methods

### addArmour

public void **addArmour**(double armour)

---

### addAttack

public void **addAttack**(double attack)

---

### addHP

public void **addHP**(double hp)

# Package pl.projekt.game.item

## Class Summary

### [AbstractItem](#)

Klasa abstrakcyjna po której dziedziczÄ... wszystkie przedmioty.

### [Armor](#)

Przedmiot dostÄ™pny dla kaÅ¼dego moba. ZwiÄ™ksza znacząco iloÅ›ć punktów defensywy oraz nieznaczÄ...co zdrowie.

### [ArmorTest](#)

### [Axe](#)

Przedmiot dostÄ™pny tylko dla Orka. ZwiÄ™ksza znacznie atak,ale zmniejsza nieznacznie defensywÄ™.

## [Dagger](#)

Przedmiot dostępny dla wszystkich. Nieznacznie zwiększa atak.

## [Hammer](#)

Przedmiot dostępny wyłącznie dla Krasnoluda.

## [Jewelery](#)

Przedmiot dostępny dla wszystkich mobów. Nieznacznie zwiększa wszystkie statystyki.

## [Maze](#)

Przedmiot dostępny wyłącznie dla Minotaura. Zwiększa on bardzo znacznie atak, za to zmniejsza nieznacznie hp oraz defensywę.

## [Shield](#)

Przedmiot dostępny dla wszystkich. Zwiększa nieznacznie zarówno hp, jak i defensywę.

## [Sword](#)

Przedmiot dostępny wyłącznie dla Elfa.

---

pl.projekt.game.item

# Class AbstractItem

```
java.lang.Object
|
+--pl.projekt.game.item.AbstractItem
```

## All Implemented Interfaces:

[IStats](#)

## Direct Known Subclasses:

[Armor](#), [Axe](#), [Dagger](#), [Hammer](#), [Jewelery](#), [Maze](#), [Shield](#), [Sword](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public abstract class AbstractItem
extends java.lang.Object
implements IStats
```

Klasa abstrakcyjna po której dziedziczą... wszystkie przedmioty.

## Constructors

## AbstractItem

```
public AbstractItem()
```

## Methods

## getArmourPoints

```
public int getArmourPoints()
```

---

## getDamagePoints

```
public int getDamagePoints()
```

---

## getHpPoints

```
public int getHpPoints()
```

---

## getMob1

```
public AbstractMonster getMob1()
```

---

## setArmourPoints

```
public void setArmourPoints(int armourPoints)
```

---

## setDamagePoints

```
public void setDamagePoints(int damagePoints)
```

---

## setHpPoints

```
public void setHpPoints(int hpPoints)
```

---

## setMob1

```
public void setMob1(AbstractMonster mob1)
```

pl.projekt.game.item

# Class Armor

```
java.lang.Object
|
+-- AbstractItem
|
+-- pl.projekt.game.item.Armor
```

All Implemented Interfaces:

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Armor
extends AbstractItem
```

Przedmiot dostÄ™pny dla kaÅ¼dego moba. ZwiÄ™ksza znaczaco iloÅ›Ä‡ punktÃ³w defensywy oraz nieznaczÄ…co zdrowie.

## Constructors

### Armor

```
public Armor(double damagePoints,
             double armourPoints,
             double hpPoints,
             AbstractMonster mob1)
```

### Armor

```
public Armor(AbstractMonster mob1)
```

## Methods

### addArmour

```
public void addArmour(double armour)
```

metoda dodaje 2% aktualnych punktÃ³w armour moba, punkty armour Itemu oraz parametr armour do punktÃ³w armour moba ktÃ³ry posiada Item

**Parameters:**

armour - przyjmuje double

---

## addAttack

```
public void addAttack(double attack)
```

---

## addHP

```
public void addHP(double hp)
```

metoda dodaje 5% aktualnych punktów hp moba, punkty hp Itemu oraz parametr hp do punktów hp moba który posiada Item

### Parameters:

hp - przyjmuje double

---

pl.projekt.game.item

## Class ArmorTest

```
java.lang.Object
|
+--pl.projekt.game.item.ArmorTest
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class ArmorTest
extends java.lang.Object
```

## Constructors

### ArmorTest

```
public ArmorTest()
```

## Methods

### addArmour

```
public void addArmour()
```

---

## addAttack

```
public void addAttack()
```

## addHP

```
public void addHP()
```

pl.projekt.game.item

## Class Axe

```
java.lang.Object
|
+-- AbstractItem
|
+-- pl.projekt.game.item.Axe
```

All Implemented Interfaces:

[IStats](#)

< [Constructors](#) > < [Methods](#) >

```
public class Axe
extends AbstractItem
```

Przedmiot dostÄ™pny tylko dla Orka.ZwiÄ™ksza znacznie atak,ale zmniejsza nieznacznie defensywÄ™.

## Constructors

### Axe

```
public Axe(double damagePoints,
           double armourPoints,
           double hpPoints,
           AbstractMonster mob1)
```

### Axe

```
public Axe(AbstractMonster mob1)
```

## Methods

## addArmour

```
public void addArmour(double armour)
```

metoda odejmuje 3% aktualnych punktów armour moba, punkty armour Itemu oraz parametr armour do punktów armour moba który posiada Item

### Parameters:

armour - przyjmuje double

---

## addAttack

```
public void addAttack(double attack)
```

metoda dodaje 7% aktualnych punktów ataku moba, punkty Damage Itemu oraz parametr attack do punktów ataku moba który posiada Item

### Parameters:

attack - przyjmuje double

---

## addHP

```
public void addHP(double hp)
```

---

pl.projekt.game.item

# Class Dagger

```
java.lang.Object
|
+-- AbstractItem
    |
    +-- pl.projekt.game.item.Dagger
```

### All Implemented Interfaces:

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Dagger
extends AbstractItem
```

Przedmiot dostępny dla wszystkich. Nieznacznie zwiększa atak.

## Constructors



## Dagger

```
public Dagger(double damagePoints,  
              double armourPoints,  
              double hpPoints,  
              AbstractMonster mob1)
```

---

## Dagger

```
public Dagger(AbstractMonster mob1)
```

## Methods

### addArmour

```
public void addArmour(double armour)
```

---

### addAttack

```
public void addAttack(double attack)
```

metoda dodaje 3% aktualnych punktów ataku moba, punkty Damage Itemu oraz parametr attack do punktów Ataku moba który posiada Item

#### Parameters:

attack - przyjmuje double

---

### addHP

```
public void addHP(double hp)
```

---

pl.projekt.game.item

# Class Hammer

```
java.lang.Object
|
+--AbstractItem
|
+--pl.projekt.game.item.Hammer
```

## All Implemented Interfaces:

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Hammer
extends AbstractItem
```

Przedmiot dostÄ™pny wyÅ,Ä...cznie dla Krasnoluda. ZwiÄ™ksza nieznacznie atak oraz Å¼ycie.

## Constructors

### Hammer

```
public Hammer(double damagePoints,
               double armourPoints,
               double hpPoints,
               AbstractMonster mob1)
```

### Hammer

```
public Hammer(AbstractMonster mob1)
```

## Methods

### addArmour

```
public void addArmour(double armour)
```

metoda dodaje punkty armour Itemu oraz parametr armour do punktÅ³w armour moba ktÅ³ry posiada Item

#### Parameters:

armour - przyjmuje double

---

## addAttack

```
public void addAttack(double attack)
```

metoda dodaje 4% aktualnych punktów ataku moba, punkty ataku Itemu oraz parametr attack do punktów Ataku moba który posiada Item

### Parameters:

attack - przyjmuje double

---

## addHP

```
public void addHP(double hp)
```

metoda dodaje 2% aktualnych punktów hp moba oraz parametr hp do punktów hp moba który posiada Item

### Parameters:

hp - przyjmuje double

---

pl.projekt.game.item

# Class Jewellery

```
java.lang.Object
|
+-- AbstractItem
|
+-- pl.projekt.game.item.Jewellery
```

### All Implemented Interfaces:

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Jewellery
extends AbstractItem
```

Przedmiot dostępny dla wszystkich mobów. Nieznacznie zwiększa wszystkie statystyki.

## Constructors

### Jewellery

```
public Jewellery(double damagePoints,
                 double armourPoints,
                 double hpPoints,
                 AbstractMonster mob)
```

---

# Jewelery

public Jewelery([AbstractMonster](#) mob1)

## Methods

### addArmour

public void **addArmour**(double armour)

metoda dodaje 2% aktualnych punktów armour moba, punkty armour Itemu oraz parametr armour do punktów armour moba który posiada Item

**Parameters:**

armour - przyjmuje double

---

### addAttack

public void **addAttack**(double attack)

metoda dodaje 2% aktualnych punktów ataku moba, punkty ataku Itemu oraz parametr attack do punktów ataku moba który posiada Item

**Parameters:**

attack - przyjmuje double

---

### addHP

public void **addHP**(double hp)

metoda dodaje 2% aktualnych punktów hp moba, punkty hp Itemu oraz parametr hp do punktów hp moba który posiada Item

**Parameters:**

hp - przyjmuje double

---

pl.projekt.game.item

# Class Maze

```
java.lang.Object
|
+--AbstractItem
|
+--pl.projekt.game.item.Maze
```

## All Implemented Interfaces:

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Maze
extends AbstractItem
```

Przedmiot dostÄ™pny wyÅ,Ä...cznie dla Minotaura.ZwiÄ™ksza on bardzo znacznie atak,za to zmiajsza nieznacznie hp oraz defensywÄ™.

## Constructors

### Maze

```
public Maze(double damagePoints,
            double armourPoints,
            double hpPoints,
            AbstractMonster mob1)
```

### Maze

```
public Maze(AbstractMonster mob1)
```

## Methods

### addArmour

```
public void addArmour(double armour)
```

metoda odejmuje 3% aktualnych punktÅ³w armour moba,punkty armour Itemu oraz parametr armour do punktÅ³w armour moba ktÅ³ry posiada Item

#### Parameters:

armour - przyjmuje armour

---

## addAttack

```
public void addAttack(double attack)
```

metoda dodaje 10% aktualnych punktów ataku moba, punkty ataku Itemu oraz parametr attack do punktów ataku moba który posiada Item

### Parameters:

attack - przyjmuje double

---

## addHP

```
public void addHP(double hp)
```

metoda odejmuje 3% aktualnych punktów hp moba, punkty hp Itemu oraz parametr hp do punktów hp moba który posiada Item

### Parameters:

hp - przyjmuje double

---

pl.projekt.game.item

# Class Shield

```
java.lang.Object
|
+-- AbstractItem
    |
    +-- pl.projekt.game.item.Shield
```

### All Implemented Interfaces:

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Shield
extends AbstractItem
```

Przedmiot dostępny dla wszystkich. Zwiększa nieznacznie zarówno hp, jak i defensywę.

## Constructors

### Shield

```
public Shield(double armourPoints,
              double damagePoints,
              double hpPoints,
              AbstractMonster mob1)
```

---

## Shield

```
public Shield(AbstractMonster mob1)
```

## Methods

### addArmour

```
public void addArmour(double armour)
```

metoda dodaje 4% aktualnych punktów armour moba, punkty armour Itemu oraz parametr armour do punktów armour moba który posiada Item

**Parameters:**

armour - przyjmuje double

---

### addAttack

```
public void addAttack(double attack)
```

---

### addHP

```
public void addHP(double hp)
```

metoda dodaje 2% aktualnych punktów hp moba, punkty hp Itemu oraz parametr hp do punktów hp moba który posiada Item

**Parameters:**

hp - przyjmuje double

---

pl.projekt.game.item

## Class Sword

```
java.lang.Object
|
+--AbstractItem
    |
    +--pl.projekt.game.item.Sword
```

**All Implemented Interfaces:**

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Sword
extends AbstractItem
```

Przedmiot dostÄ™pny wyÅ„Ä…cznie dla Elfa. ZwiÄ™ksza znacznie atak moba.

## Constructors

### Sword

```
public Sword(double armourPoints,
             double damagePoints,
             double hpPoints,
             AbstractMonster mob1)
```

### Sword

```
public Sword(AbstractMonster mob1)
```

## Methods

### addArmour

```
public void addArmour(double armour)
```

### addAttack

```
public void addAttack(double attack)
```

metoda dodaje 5% aktualnych punktÅ³w ataku moba, punkty ataku Itemu oraz parametr attack do punktÅ³w ataku moba ktÅ³ry posiada Item

**Parameters:**

attack - przyjmuje double

### addHP

```
public void addHP(double hp)
```



# Package pl.projekt.game.material

## Class Summary

### [AbstractMaterials](#)

Klasa abstrakcyjna po której dziedziczą... wszystkie materiały.

### [Diamond](#)

Materiał, dostępny dla każdego moba.

### [Iron](#)

Materiał, który może zostać zebrany przez Elfa oraz Minotaura.

### [Stone](#)

Materiał, który może być zebrany przez Krasnoluda, Minotura oraz Orka.

### [Wood](#)

Materiał, dostępny dla każdego moba. Można na dzieńki niemu stworzyć miętę kałdy przedmiot w symulacji.

pl.projekt.game.material

## Class AbstractMaterials

```
java.lang.Object
|
+--pl.projekt.game.material.AbstractMaterials
```

Direct Known Subclasses:

[Diamond](#), [Iron](#), [Stone](#), [Wood](#)

< [Constructors](#) > < [Methods](#) >

```
public abstract class AbstractMaterials
extends java.lang.Object
```

Klasa abstrakcyjna po której dziedziczą... wszystkie materiały. Znajduje się tutaj metoda która sprawdza czy materiał, nie jest za ciężki dla moba.

## Constructors

### AbstractMaterials

```
public AbstractMaterials()
```

## Methods

## getWeight

```
public abstract int getWeight()
```

---

## isNotToHeavy

```
public boolean isNotToHeavy(int weight,  
                             int maxWeight)
```

Metoda sprawdza czy w ekwipunku mob nie ma za duzo przedmiotow 1 rodzaju

### Parameters:

weight - waga przedmiotu

maxWeight - maksymalna waga ktora mob moze przeniesc

### Returns:

zwraca prawde lub falsz w zaleznosci czy mob moze podniesc wiecej czy nie

---

pl.projekt.game.material

## Class Diamond

```
java.lang.Object  
|  
+--AbstractMaterials  
|  
+--pl.projekt.game.material.Diamond
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Diamond  
extends AbstractMaterials
```

Materia, dostÄ™pny dla kaÅ¼dego moba. Potrzebny do wytworzenia BiÅ¼uterii.

## Constructors

### Diamond

```
public Diamond()
```

---

# Diamond

```
public Diamond(int weight)
```

## Methods

### getWeight

```
public int getWeight()
```

Overrides:

[getWeight](#) in class [AbstractMaterials](#)

---

pl.projekt.game.material

## Class Iron

```
java.lang.Object
|
+-- AbstractMaterials
|
+-- pl.projekt.game.material.Iron
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Iron
extends AbstractMaterials
```

Materiał, który może zostać zebrany przez Elfa oraz Minotaura. Służy do stworzenia Buzdyganu oraz Miecza.

## Constructors

### Iron

```
public Iron()
```

---

### Iron

```
public Iron(int weight)
```

## Methods

### getWeight

```
public int getWeight()
```

Overrides:

[getWeight](#) in class [AbstractMaterials](#)

---

pl.projekt.game.material

## Class Stone

```
java.lang.Object
|
+-- AbstractMaterials
|
+-- pl.projekt.game.material.Stone
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Stone
extends AbstractMaterials
```

MateriaÅ, ktÅ³ry moÅ¼e byÅ† zebrany przez Krasnoluda,Minotura oraz Orka. SÅ, uÅ¼y do stworzenia MÅ,ota,Buzdyganu oraz Topora.

## Constructors

### Stone

```
public Stone()
```

### Stone

```
public Stone(int weight)
```

## Methods

## getWeight

```
public int getWeight()
```

Overrides:

[getWeight](#) in class [AbstractMaterials](#)

---

pl.projekt.game.material

## Class Wood

```
java.lang.Object
|
+--AbstractMaterials
|
+--pl.projekt.game.material.Wood
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Wood
extends AbstractMaterials
```

Materiał, dostępny dla każdego moba. Można dzięki niemu stworzyć niemal każdy przedmiot w symulacji.

## Constructors

### Wood

```
public Wood()
```

### Wood

```
public Wood(int weight)
```

## Methods

# getWeight

```
public int getWeight()
```

## Overrides:

[getWeight](#) in class [AbstractMaterials](#)

# Package pl.projekt.game.mob

## Class Summary

### [AbstractMonster](#)

Klasa abstrakcyjna po której dziedziczy każdy mob. Dzięki niej potrafi tworzyć sztylet, zbroje czy tarcze oraz zbierać diamenty i drewno. Znajdują się w niej metody na zbieranie materiałów, tworzenie przedmiotów oraz walki i ich czenie mobów.

### [Dwarf](#)

Krasnolud jest jednym z 4 ras pojawiających się w symulacji.

### [Elf](#)

Elf jest jednym z 4 ras pojawiających się w symulacji.

### [Minotaur](#)

Minotaur jest jednym z 4 ras pojawiających się w symulacji.

### [MinotaurTest](#)

### [Orc](#)

Ork jest jednym z 4 ras pojawiających się w symulacji.

---

pl.projekt.game.mob

## Class AbstractMonster

```
java.lang.Object
|
+--pl.projekt.game.mob.AbstractMonster
```

All Implemented Interfaces:

[IStats](#)

Direct Known Subclasses:

[Dwarf](#), [Elf](#), [Minotaur](#), [Orc](#)

---

< [Constructors](#) > < [Methods](#) >

---

public abstract class **AbstractMonster**

extends java.lang.Object

implements [IStats](#)

Klasa abstrakcyjna po której dziedziczy każdy mob. Dzięki niej potrafi tworzyć sztylet, zbroje czy tarcze oraz zbierać diamenty i drewno. Znajdują się w niej metody na zbieranie materiałów, tworzenie przedmiotów oraz walki i ich czenie mobów.

## Constructors

# AbstractMonster

```
public AbstractMonster()
```

## Methods

### addArmour

```
public void addArmour(double Armr)
```

---

### addAttack

```
public void addAttack(double Attack)
```

---

### addHP

```
public void addHP(double HP)
```

---

### collectMaterial

```
public void collectMaterial(AbstractMonster monster1,  
                             AbstractMaterials materials)
```

Metoda sprawdza czy materiaÅ, moÅ¼e zostaÅ± zebrany przez moba oraz zmniejsza posiadane przez moba wÅne miejsce na materiaÅ,y

**Parameters:**

monster1 - mob ktÅ³ry zbiera materiaÅ,  
materials - zbierany materiaÅ,

---

### craftNewItem

```
public void craftNewItem(AbstractMonster monster)
```

metoda w ktÅ³rej mob wybiera jaki item ma stworzyÅ±

**Parameters:**

monster - mob ktÅ³ry tworzy itemter

---



## fight

```
public AbstractMonster fight(AbstractMonster firstM,  
                               AbstractMonster secondM)
```

Metoda wykonuje walke 2 mobow roznych klas a nastepnie zwraca wygranego

### Parameters:

firstM - 1 mob  
secondM - 2 mob

### Returns:

zwraca wygranego moba

---

## getAttack

```
public double getAttack()
```

---

## getDefence

```
public double getDefence()
```

---

## getEquipment

```
public java.util.ArrayList getEquipment()
```

---

## getHealth

```
public double getHealth()
```

---

## getWoodnmb

```
public int getWoodnmb()
```

---

## merge

```
public AbstractMonster merge(AbstractMonster monster1,  
                                AbstractMonster monster2)
```

Metoda ktora w parametrach przyjmuje 2 moby tej samej klasy a nastepnie laczy je w 1 nowego moba o sumie parametrow

### Parameters:

monster1 - 1 mob  
monster2 - 2 mob

### Returns:

zwraca nowego moba

---

pl.projekt.game.mob

## Class Dwarf

```
java.lang.Object  
|  
+--AbstractMonster  
|  
+--pl.projekt.game.mob.Dwarf
```

### All Implemented Interfaces:

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Dwarf  
extends AbstractMonster
```

Krasnolud jest jednym z 4 ras pojawiajÄ...cych siÄ™ w symulacji. Charakteryzuje siÄ™ wysokÄ...  
obronÄ..., Ärednim Ä¼yciem i niskim atakiem. Posiada moÄ¼liwoÄ¸ tworzenia MÄota.

## Constructors

### Dwarf

```
public Dwarf()
```

---

### Dwarf

```
public Dwarf(double healthPoints,  
              double defencePoints,  
              double attacPoints)
```

## Methods

### addArmour

```
public void addArmour(double armr)
```

**Overrides:**

[addArmour](#) in class [AbstractMonster](#)

---

### addAttack

```
public void addAttack(double attack)
```

**Overrides:**

[addAttack](#) in class [AbstractMonster](#)

---

### addHP

```
public void addHP(double HP)
```

**Overrides:**

[addHP](#) in class [AbstractMonster](#)

---

### collectStone

```
public void collectStone()
```

metoda zwiększa ilość posiadanego przez moba kamienia o 1

---

### createHammer

```
public void createHammer()
```

metoda tworzy oraz dodaje do Ekwipunku moba jedną... sztukę Młota oraz odejmuje od ilości kawałków drewna oraz kamienia cenę za stworzenie Młota (odpowiednio 2 i 1)

---

## getAttack

```
public double getAttack()
```

**Overrides:**

[getAttack](#) in class [AbstractMonster](#)

---

## getDefence

```
public double getDefence()
```

**Overrides:**

[getDefence](#) in class [AbstractMonster](#)

---

## getHealth

```
public double getHealth()
```

**Overrides:**

[getHealth](#) in class [AbstractMonster](#)

---

pl.projekt.game.mob

## Class Elf

```
java.lang.Object
|
+--AbstractMonster
|
+--pl.projekt.game.mob.Elf
```

**All Implemented Interfaces:**

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Elf
extends AbstractMonster
```

Elf jest jednym z 4 ras pojawiających się w symulacji. Charakteryzuje się standardowymi statystykami oraz możliwością stworzenia Miecza.

## Constructors

## Elf

```
public Elf()
```

---

## Elf

```
public Elf(double healthPoints,  
           double defencePoints,  
           double attacPoints)
```

## Methods

### addArmour

```
public void addArmour(double armr)
```

Overrides:

[addArmour](#) in class [AbstractMonster](#)

---

### addAttack

```
public void addAttack(double attack)
```

Overrides:

[addAttack](#) in class [AbstractMonster](#)

---

### addHP

```
public void addHP(double HP)
```

Overrides:

[addHP](#) in class [AbstractMonster](#)

---

### collectIron

```
public void collectIron()
```

Zwiększa ilość posiadanego przez moba żelaza o 1

---

## createSword

```
public void createSword()
```

metoda tworzy oraz dodaje do Ekwipunku moba jedn... sztuk<sup>™</sup> Miecza oraz odejmuje od iloŹci ¼elaza oraz kawałk<sup>3</sup>w drewna cen<sup>™</sup> za stworzenie Miecza (Odpowiednio 2 i 1)

---

## getAttack

```
public double getAttack()
```

Overrides:

[getAttack](#) in class [AbstractMonster](#)

---

## getDefence

```
public double getDefence()
```

Overrides:

[getDefence](#) in class [AbstractMonster](#)

---

## getHealth

```
public double getHealth()
```

Overrides:

[getHealth](#) in class [AbstractMonster](#)

---

pl.projekt.game.mob

# Class Minotaur

```
java.lang.Object
|
+--AbstractMonster
    |
    +--pl.projekt.game.mob.Minotaur
```

All Implemented Interfaces:

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Minotaur
```

extends [AbstractMonster](#)

Minotaur jest jednym z 4 ras pojawiających się™ w symulacji. Charakteryzuje się™ wysokim atakiem oraz niskim 1/4yciem oraz obroną....Potrafi wytorzyć Buzdygan.

## Constructors

### Minotaur

```
public Minotaur()
```

### Minotaur

```
public Minotaur(double healthPoints,  
                double defencePoints,  
                double attacPoints)
```

## Methods

### addArmour

```
public void addArmour(double armr)
```

Overrides:

[addArmour](#) in class [AbstractMonster](#)

### addAttack

```
public void addAttack(double attack)
```

Overrides:

[addAttack](#) in class [AbstractMonster](#)

### addHP

```
public void addHP(double HP)
```

Overrides:

[addHP](#) in class [AbstractMonster](#)

## collectIron

```
public void collectIron()
```

Zwiększa ilość posiadanego przez moba żelaza o 1

---

## collectStone

```
public void collectStone()
```

metoda zwiększa ilość posiadanego przez moba kamienia o 1

---

## createMaze

```
public void createMaze()
```

metoda tworzy oraz dodaje do Ekwipunku moba jedną sztukę Buzdyganu oraz odejmuje od ilości żelaza, kawy, drewna oraz kamienia cenę za stworzenie Buzdyganu (odpowiednio 1, 1 i 2)

---

## getAttack

```
public double getAttack()
```

Overrides:

[getAttack](#) in class [AbstractMonster](#)

---

## getDefence

```
public double getDefence()
```

Overrides:

[getDefence](#) in class [AbstractMonster](#)

---

## getHealth

```
public double getHealth()
```

Overrides:

[getHealth](#) in class [AbstractMonster](#)

---



pl.projekt.game.mob

# Class MinotaurTest

```
java.lang.Object
|
+--pl.projekt.game.mob.MinotaurTest
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class MinotaurTest
extends java.lang.Object
```

## Constructors

### MinotaurTest

```
public MinotaurTest()
```

## Methods

### collectWood

```
public void collectWood()
```

---

### createJewelery

```
public void createJewelery()
```

---

pl.projekt.game.mob

# Class Orc

```
java.lang.Object
|
+--AbstractMonster
    |
    +--pl.projekt.game.mob.Orc
```

All Implemented Interfaces:

[IStats](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Orc
extends AbstractMonster
```

Ork jest jednym z 4 ras pojawiających się™ w symulacji. Charakteryzuje się™ dużym zdrowiem oraz średnim atakiem oraz defensywą....Potrafi tworzyć Top³r.

## Constructors

### Orc

```
public Orc()
```

### Orc

```
public Orc(double healthPoints,
           double defencePoints,
           double attacPoints)
```

## Methods

### addArmour

```
public void addArmour(double armr)
```

**Overrides:**

[addArmour](#) in class [AbstractMonster](#)

### addAttack

```
public void addAttack(double attack)
```

**Overrides:**

[addAttack](#) in class [AbstractMonster](#)

---

## addHP

```
public void addHP(double HP)
```

**Overrides:**

[addHP](#) in class [AbstractMonster](#)

---

## collectStone

```
public void collectStone()
```

metoda zwiÄ™ksza iloÅ›Ä™ posiadanego przez moba kamienia o 1

---

## createAxe

```
public void createAxe()
```

metoda tworzy oraz dodaje do Ekwipunku moba jednÄ… sztukÄ™ Toporu oraz odejmuje od iloÅ›ci kawaÅ³kÃ³w drewna oraz kamienia cenÄ™ za stworzenie Toporu(odpowiednio 2 i 2)

---

## getAttack

```
public double getAttack()
```

**Overrides:**

[getAttack](#) in class [AbstractMonster](#)

---

## getDefence

```
public double getDefence()
```

**Overrides:**

[getDefence](#) in class [AbstractMonster](#)

---

## getHealth

```
public double getHealth()
```

**Overrides:**

[getHealth](#) in class [AbstractMonster](#)

# Package pl.projekt.simulation

## Interface Summary

[IRandom](#)

## Class Summary

[Board](#)

Klasa, na kt<sup>3</sup>rej obiekcie pojawiaj<sup>3</sup> si<sup>TM</sup> materia<sup>3</sup>,y oraz moby walcz<sup>3</sup>.../<sup>3</sup>,<sup>3</sup>...cz<sup>3</sup>... si<sup>TM</sup>

[SimulationApp](#)

Glowna klasa programu posiada metode main oraz obsluguje inne klasy

pl.projekt.simulation

## Class Board

```
java.lang.Object
|
+--pl.projekt.simulation.Board
```

All Implemented Interfaces:

[IRandom](#)

< [Constructors](#) > < [Methods](#) >

```
public class Board
extends java.lang.Object
implements IRandom
```

Klasa, na kt<sup>3</sup>rej obiekcie pojawiaj<sup>3</sup> si<sup>TM</sup> materia<sup>3</sup>,y oraz moby walcz<sup>3</sup>.../<sup>3</sup>,<sup>3</sup>...cz<sup>3</sup>... si<sup>TM</sup>

## Constructors

### Board

```
public Board(int size,
             int mobs)
```

## Methods

## getInfo

```
public java.lang.String getInfo()
```

Metoda tworzy nowego stringa z informacjami o ilosci mobow po rundzie

**Returns:**

zwraca String z informacjami o mobach

---

## getPositionX

```
public int getPositionX()
```

Metoda losuje nowa kordynate X

**Returns:**

zwraca kordynate X

---

## getPositionY

```
public int getPositionY()
```

Metoda losuje nowa kordynate Y

**Returns:**

zwraca kordynate Y

---

## move

```
public void move()
```

Metoda uzywajac metody lookForMobsszuka miejsc na ktorych wystepuja moby nastepnie uzywajac w petli uzywajac metod getPositionX i getPositionY losuje nowe kordynaty dla wybranego moba nastepnie sprawdza czy nowe kordynaty sa puste czy wystepuje na nich jakis mob/metrial jesli na na nowych kordyjatach znajduje sie material uzywa metody collectMaterial jesli na nowych kordynatach znajduje sie mob takiej samej klasy uzywa metody merge jesli na nowych kordynatach znajduje sie mob innej klasy uzywa metody fight

---

## onlyOneSpeciesLeft

```
public boolean onlyOneSpeciesLeft()
```

Metoda sprawdza czy na planszy znajduje sie tylko jeden gatunek moba

**Returns:**

zwraca prawde jesli znajduje sie tylko 1 a falsz jesli kilka

---

## placeOnTheBoard

```
public void placeOnTheBoard()
```

Metoda tworzy nowa plansze a nastepnie ustawia na niej moby i materialy

---

## setMaterialPosition

```
public void setMaterialPosition()
```

Metoda tworzy nowe materialy i ustawia je w randomowych miejscach na planszy

---

## setMobPosition

```
public void setMobPosition()
```

Metoda tworzy ilosc mobow podanych przez uzytkownika a nastepnie przy pomocy ustawia w randomowych miejscach na planszy

---

pl.projekt.simulation

# Interface IRandom

---

< [Methods](#) >

---

```
public interface IRandom
```

---

## Methods

### getPositionX

```
public int getPositionX()
```

---

### getPositionY

```
public int getPositionY()
```

---

## move

```
public void move()
```

---

## setMaterialPosition

```
public void setMaterialPosition()
```

---

## setMobPosition

```
public void setMobPosition()
```

---

pl.projekt.simulation

# Class SimulationApp

```
java.lang.Object
|
+--pl.projekt.simulation.SimulationApp
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class SimulationApp
extends java.lang.Object
```

Główna klasa programu posiada metode main oraz obsługuje inne klasy

### Version:

v0.9.1

### Author:

Jakub Gliwa, Kacper Ziejlo

---

## Constructors

### SimulationApp

```
public SimulationApp()
```

---

## Methods

# main

```
public static void main(java.lang.String[] args)
```



# INDEX

## A

[addArmour](#) ... 2  
[addArmour](#) ... 5  
[addArmour](#) ... 6  
[addArmour](#) ... 8  
[addArmour](#) ... 9  
[addArmour](#) ... 10  
[addArmour](#) ... 12  
[addArmour](#) ... 13  
[addArmour](#) ... 15  
[addArmour](#) ... 16  
[addArmour](#) ... 24  
[addArmour](#) ... 27  
[addArmour](#) ... 29  
[addArmour](#) ... 31  
[addArmour](#) ... 34  
[addAttack](#) ... 2  
[addAttack](#) ... 6  
[addAttack](#) ... 7  
[addAttack](#) ... 8  
[addAttack](#) ... 9  
[addAttack](#) ... 11  
[addAttack](#) ... 12  
[addAttack](#) ... 14  
[addAttack](#) ... 15  
[addAttack](#) ... 16  
[addAttack](#) ... 24  
[addAttack](#) ... 27  
[addAttack](#) ... 29  
[addAttack](#) ... 31  
[addAttack](#) ... 34  
[addHP](#) ... 2  
[addHP](#) ... 6  
[addHP](#) ... 7  
[addHP](#) ... 8  
[addHP](#) ... 9  
[addHP](#) ... 11  
[addHP](#) ... 12  
[addHP](#) ... 14  
[addHP](#) ... 15  
[addHP](#) ... 16  
[addHP](#) ... 24  
[addHP](#) ... 27  
[addHP](#) ... 29  
[addHP](#) ... 31  
[addHP](#) ... 35  
[AbstractItem](#) ... 3  
[AbstractItem](#) ... 3  
[AbstractMaterials](#) ... 17  
[AbstractMaterials](#) ... 17  
[AbstractMonster](#) ... 23  
[AbstractMonster](#) ... 24  
[Armor](#) ... 5  
[Armor](#) ... 5  
[Armor](#) ... 5  
[ArmorTest](#) ... 6  
[ArmorTest](#) ... 6  
[Axe](#) ... 7  
[Axe](#) ... 7  
[Axe](#) ... 7

## B

[Board](#) ... 36  
[Board](#) ... 36

## C

[collectIron](#) ... 29  
[collectIron](#) ... 32  
[collectMaterial](#) ... 24  
[collectStone](#) ... 27  
[collectStone](#) ... 32  
[collectStone](#) ... 35  
[collectWood](#) ... 33  
[craftNewItem](#) ... 24  
[createAxe](#) ... 35  
[createHammer](#) ... 27  
[createJewelry](#) ... 33  
[createMaze](#) ... 32  
[createSword](#) ... 30

## D

[Dagger](#) ... 8  
[Dagger](#) ... 9  
[Dagger](#) ... 9  
[Diamond](#) ... 18  
[Diamond](#) ... 18  
[Diamond](#) ... 19  
[Dwarf](#) ... 26  
[Dwarf](#) ... 26  
[Dwarf](#) ... 26

## E

[Elf](#) ... 28  
[Elf](#) ... 29  
[Elf](#) ... 29

## F

[fight](#) ... 25

## G

[getArmourPoints](#) ... 4  
[getAttack](#) ... 25  
[getAttack](#) ... 28  
[getAttack](#) ... 30  
[getAttack](#) ... 32  
[getAttack](#) ... 35  
[getDamagePoints](#) ... 4  
[getDefence](#) ... 25  
[getDefence](#) ... 28  
[getDefence](#) ... 30  
[getDefence](#) ... 32  
[getDefence](#) ... 35  
[getEquipment](#) ... 25  
[getHealth](#) ... 25  
[getHealth](#) ... 28  
[getHealth](#) ... 30  
[getHealth](#) ... 32  
[getHealth](#) ... 35  
[getHpPoints](#) ... 4  
[getInfo](#) ... 37  
[getMob1](#) ... 4  
[getPositionX](#) ... 37  
[getPositionX](#) ... 38  
[getPositionY](#) ... 37  
[getPositionY](#) ... 38  
[getWeight](#) ... 18  
[getWeight](#) ... 19  
[getWeight](#) ... 20  
[getWeight](#) ... 21  
[getWeight](#) ... 22  
[getWoodnmb](#) ... 25

## H

[Hammer](#) ... 10  
[Hammer](#) ... 10  
[Hammer](#) ... 10

## I

[isNotToHeavy](#) ... 18  
[IRandom](#) ... 38  
[Iron](#) ... 19  
[Iron](#) ... 19  
[Iron](#) ... 19  
[IStats](#) ... 2

## J

[Jewelery](#) ... 11  
[Jewelery](#) ... 11  
[Jewelery](#) ... 12

## M

[main](#) ... 40  
[merge](#) ... 26  
[move](#) ... 37  
[move](#) ... 39  
[Maze](#) ... 13  
[Maze](#) ... 13  
[Maze](#) ... 13  
[Minotaur](#) ... 30  
[Minotaur](#) ... 31  
[Minotaur](#) ... 31  
[MinotaurTest](#) ... 33  
[MinotaurTest](#) ... 33

## O

[onlyOneSpeciesLeft](#) ... 37  
[Orc](#) ... 33  
[Orc](#) ... 34  
[Orc](#) ... 34

## P

[placeOnTheBoard](#) ... 38

## S

[setArmourPoints](#) ... 4  
[setDamagePoints](#) ... 4  
[setHpPoints](#) ... 4  
[setMaterialPosition](#) ... 38  
[setMaterialPosition](#) ... 39  
[setMob1](#) ... 4  
[setMobPosition](#) ... 38  
[setMobPosition](#) ... 39  
[Shield](#) ... 14  
[Shield](#) ... 14  
[Shield](#) ... 15  
[SimulationApp](#) ... 39  
[SimulationApp](#) ... 39  
[Stone](#) ... 20  
[Stone](#) ... 20  
[Stone](#) ... 20  
[Sword](#) ... 15  
[Sword](#) ... 16  
[Sword](#) ... 16

## W

[Wood](#) ... 21  
[Wood](#) ... 21  
[Wood](#) ... 21