



Faculty of Software Engineering and Computer Systems

Programming

Lecture #2.
Introduction to OOP.

Instructor of faculty
Pismak Alexey Evgenievich
Kronverksky Pr. 49, 374 room
pismak@itmo.ru

Saint-Petersburg

ООП... Что это и зачем?

- Процедурное программирование
- Автоматное программирование
- Событийно-ориентированное программирование
- Логическое программирование
- Функциональное программирование
- Объектно-ориентированное программирование

Objects of real world...

...for example, humans

```
public class Main {
```

```
    public static void main(String args[]) {
```

```
        String name = "Шустрый";           // имя
        int health = 625;                    // здоровье
        int mana = 820;                      // авторитет
        int age = 28;                        // возраст
        long longitude = ...;                // координаты местоположения
        long latitude = ...;                 // долгота и широта
```

```
    }
}
```

Objects of real world can...

```
public class Main {  
  
    // main пропущен  
  
    public static void sayPhrase() {  
        System.out.println("Самые быстрые ноги на районе");  
    }  
  
    public static long runToWest(int age, int heath) {  
        speed = health * age;  
        return longitude - speed;  
    }  
}
```

Objects of real world can...

```
public class Main {
```

```
    public static void main(String args[]) {
```

```
        // объявление переменных пропущено
```

```
        sayPhrase();
```

```
        if( runToWest(age, health) == 30_375399) { // в Купчино
```

```
            mana++;
```

```
        }
```

```
    }
```

```
}
```

What if there are two of them?

```
String name = "Шустрый";           // имя
int health = 625;                   // здоровье
int mana = 820;                     // авторитет
int age = 21;                       // возраст
long longitude = ...;               // координаты местоположения
long latitude = ...;               // долгота и широта

String name2 = "Тормоз";           // имя_второго
int health2 = 400;                  // здоровье_второго
int mana2 = 747;                    // авторитет_второго
int age2 = 24;                      // возраст_второго
long longitude = ...;               // координаты местоположения
long latitude = ...;               // долгота и широта
```

What if there are two of them?

```
// тут алгоритм для каждого разный  
public static void sayPhrase() {  
    System.out.println("Семки есть?");  
}
```

```
// тут алгоритм одинаковый  
public static long runToWest(int age, int heath) {  
    speed = health * age;  
    return longitude - speed;  
}
```

What if there are two of them?

```
public class Main {

    public static void main(String args[]) {

        String name = "Шустрый";           // имя
        int health = 625;                   // здоровье
        int mana = 820;                     // авторитет
        int age = 21;                       // возраст
        long longitude = ...;               // координаты местоположения
        long latitude = ...;                // долгота и широта

        String name2 = "Тормоз";           // имя_второго
        int health2 = 400;                   // здоровье_второго
        int mana2 = 747;                     // авторитет_второго
        int age2 = 24;                       // возраст_второго
        long longitude2 = ...;               // координаты местоположения
        long latitude2 = ...;                // долгота и широта

        sayPhrase1();
        if( runToWest(age, health) == runToWest(age2, health2)) { // встретились
            sayPhrase2();
        }
    }

    public static void sayPhrase1() {
        System.out.println("Самые быстрые ноги на районе");
    }
    public static void sayPhrase2() {
        System.out.println("Семки есть?");
    }
    public static long runToWest(int age, int heath) {
        speed = health * age;
        return longitude - speed;
    }
}
```


What if there are two of them?

```
public class Main {  
  
    public static void main(String args[]) {  
  
        String name = "Шустрый";  
        int health = 625;  
        int mana = 820;  
        int age = 21;  
        long longitude = ...;  
        long latitude = ...;  
  
        String name2 = "Тормоз";  
        int health2 = 400;  
        int mana2 = 747;  
        int age2 = 24;  
        long longitude2 = ...;  
        long latitude2 = ...;  
  
        sayPhrase1();  
        if( runToWest(age, health) == runToWest(age2, health2)) { // встретились  
            sayPhrase2();  
        }  
    }  
  
    public static void sayPhrase1() {  
        System.out.println("Самые быстрые ноги на районе");  
    }  
    public static void sayPhrase2() {  
        System.out.println("Семки есть?");  
    }  
    public static long runToWest(int age, int heath) {  
        speed = health * age;  
        return longitude - speed;  
    }  
}
```

А что если их много?

First approach

```
String name[] = { ... };           // имена
int health[] = ...;                // здоровье
int mana[] = ...;                  // авторитеты
int age[] = ...;                   // возраста
long longitude[] = ...;            // координаты местоположения
long latitude[] = ...;             // долгота и широта

public static void sayPhrase1() { }
public static void sayPhrase2() { }
// ...
public static void sayPhraseN() { }
```

А что, если у каждого несколько имен?

Second approach

// собственный тип данных

class Guy {

String name;

// имя

int health;

// здоровье

int mana;

// авторитет

int age;

// возраст

long longitude;

// координаты местоположения

long latitude;

// долгота и широта

}

// Терминология:

// класс

// Файл для класса

// поле

// Значения по умолчанию

Class contains properties and actions

// собственный тип данных

class Guy {

// объявление остальных полей опущено

String message = "Самые быстрые ноги на районе";

public void sayPhrase() { // методы без static

System.out.println(message);

}

public long runToWest() {

speed = health * age;

return longitude - speed;

}

}

How use it?

```
1. public static void main(String args[]) {  
2.     // экземпляр класса  
3.     Guy guy = new Guy();  
4.     guy.health = 1; // обращение к атрибутам (св-вам)  
5.     int guyMana = guy.mana;  
6.     guy.sayPhrase ();  
7.     if ( guy.runToWest() > 0 ) {  
8.         System.out.println("Hey, guys");  
9.     }  
10.}
```

How use it?

```
1. public static void main(String args[]) {  
2.     Guy faster = new Guy();  
3.     Guy slower = new Guy();  
4.     slower.sayPhrase ();  
5.     if ( faster.runToWest() == slower.runToWest() ) {  
6.         System.out.println("Hey, guy");  
7.     }  
8. }
```

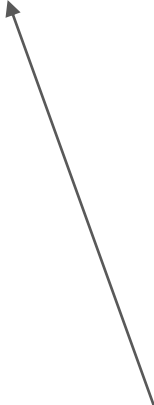
Creation of class instances

```
Guy guy = new Guy();
```

Classes & Objects

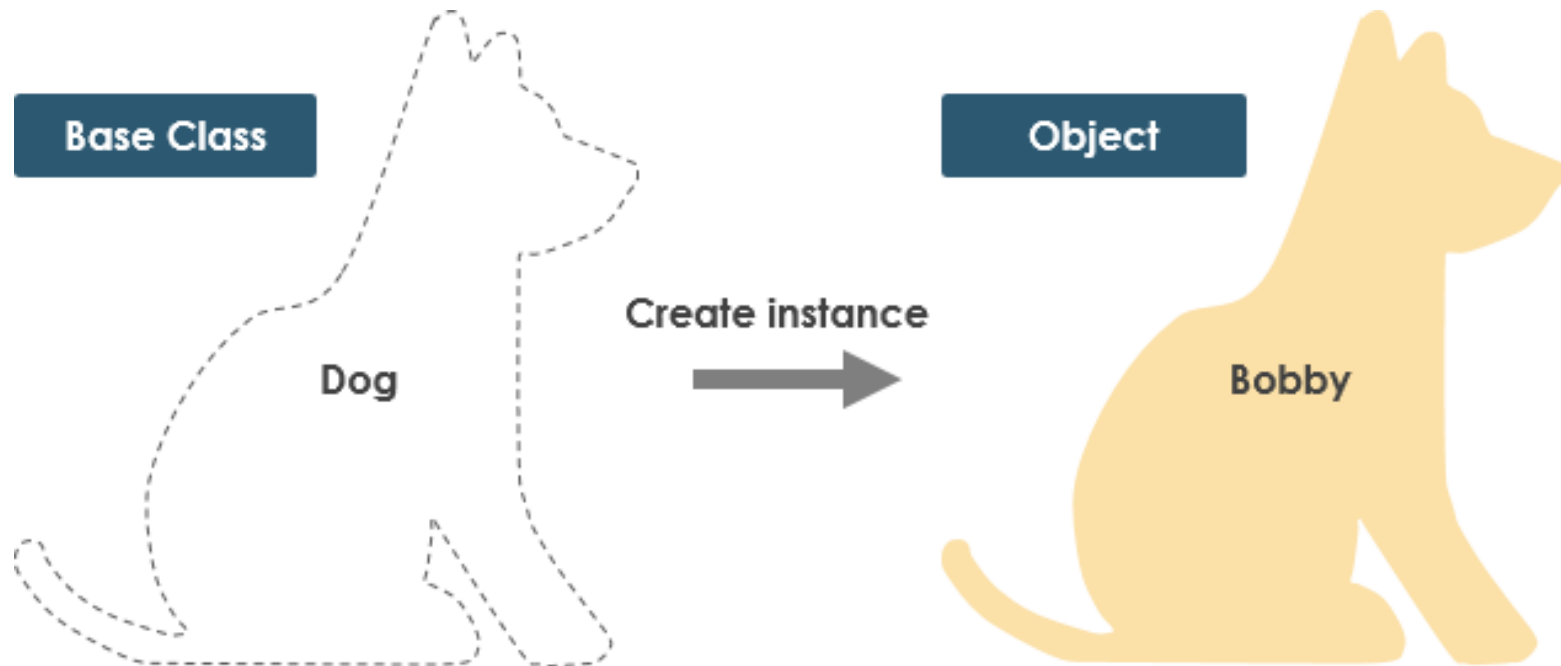
```
class Guy {  
    String name;  
    int health;  
    int mana;  
    int age;  
}
```

```
class Main {  
    public static void main(String args[]) {  
  
        Guy guy = new Guy();  
  
    }  
}
```



Ссылка на объект типа
Guy, где Guy - это класс

Dog bobby = new Dog();



Properties

Color

Eye Color

Height

Length

Weight

Methods

Sit

Lay Down

Shake

Come

Property Values

Color: Yellow

Eye Color: Brown

Height: 17 in

Length: 35 in

Weight: 24 pounds

Methods

Sit

Lay Down

Shake

Come

Constructors

```
Guy guy = new Guy();
```

```
new Guy(); // new выделяет под объект память и... Guy()
```



Что это такое и для чего?

Constructors

```
class Guy {
```

```
    String name;           // имя - null
    int health;            // здоровье - 0
    int mana;              // авторитет - 0
    int age;               // возраст - 0
```

```
    ...
```

```
    public Guy() {
        name = "Шустрый";
    }
```

```
    public Guy(String newName) {
        name = newName;
    }
}
```

```
Guy guy = new Guy();
guy.name; // "Шустрый"
```

```
Guy guy = new Guy("Жидкий");
guy.name; // Жидкий
```

Constructors

```
class Guy {
```

```
    String name;           // имя - null  
    int health;            // здоровье - 0  
    int mana;              // авторитет - 0  
    int age;               // возраст - 0  
    ...
```

```
    public Guy(String newName) {  
        name = newName;  
    }  
}
```

```
Guy guy = new Guy(); // Error
```

Constructors

```
class Guy {
```

```
    String name;
```

```
    int health;
```

```
    int mana;
```

```
    int age;
```

```
    ...
```

```
    // нет конструкторов
```

```
}
```

```
// имя - null
```

```
// здоровье - 0
```

```
// авторитет - 0
```

```
// возраст - 0
```

```
Guy guy = new Guy(); // OK
```

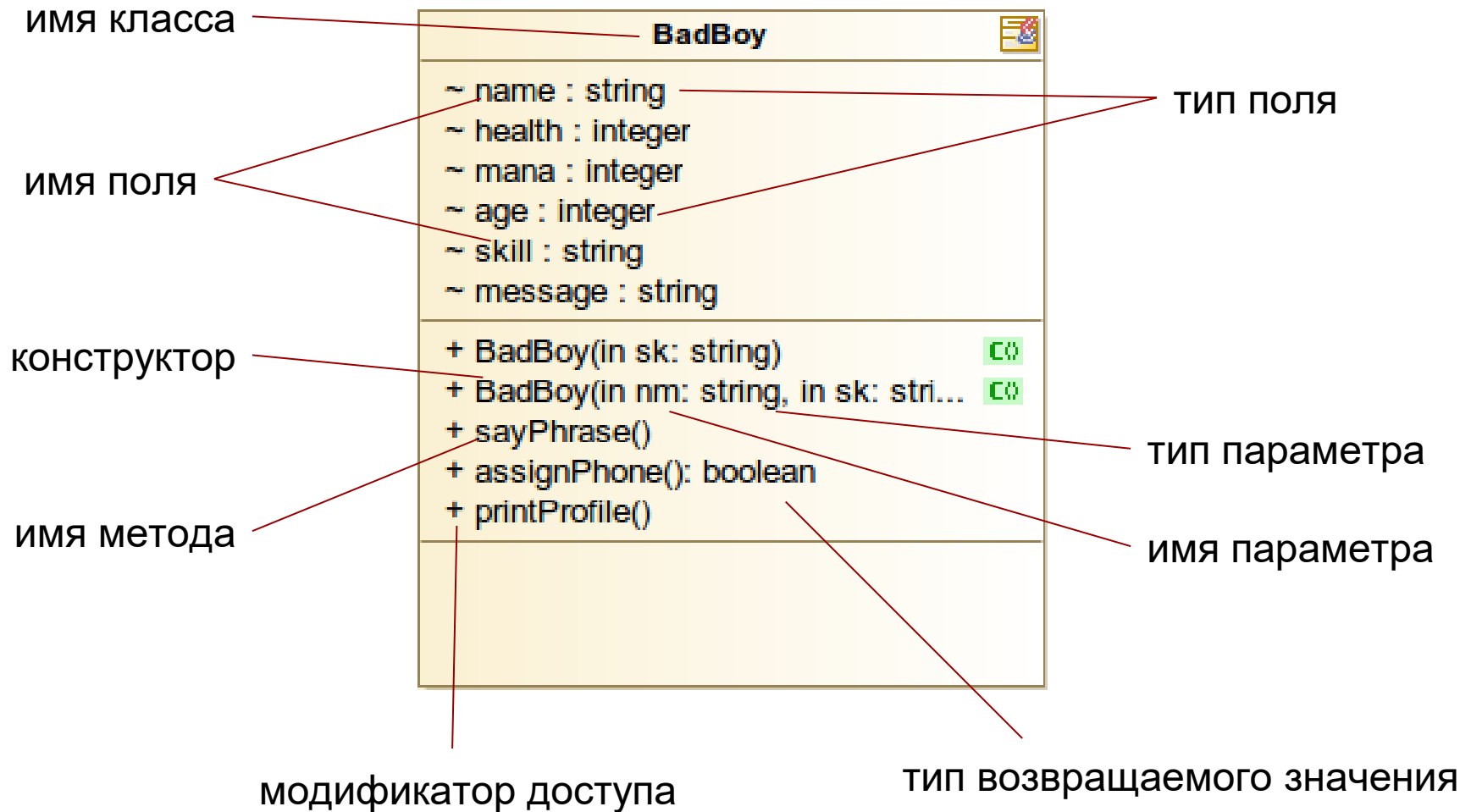
```
Guy guy = new Guy("Boy"); // Error
```

Class description in UML

Немного википедии

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения.

Class description in UML



Using your custom classes

```
import java.util.Random;  
import Guy ?
```

```
public class Main {  
    public static void main (String [] args) {  
  
        Random random = new Random();  
        Guy guy = new Guy();  
  
    }  
}
```

01) Для чего вообще писать import?

10) Откуда берутся названия java.lang, java.util и другие?

Packages

```
package ru.ifmo.se.prog.examples;
```

```
public class Guy {
```

```
    // все поля и методы класса
```

```
}
```

**Какие есть правила именования пакетов?
Влияет ли на класс (файл) наличие пакета?**

Packages

```
import ru.ifmo.se.prog.examples.Guy;  
  
public class Main {  
  
    public static void main ( String[] args ) {  
  
        Guy boy = new Guy();  
  
        // ...  
    }  
}
```

А класс Main... он в каком пакете?

Packages

```
package ru.ifmo.se.prog;
```

```
import ru.ifmo.se.prog.examples.BadBoy;
```

```
public class Main {
```

```
    public static void main ( String[] args ) {
```

```
        BadBoy veryBadBoy = new BadBoy();
```

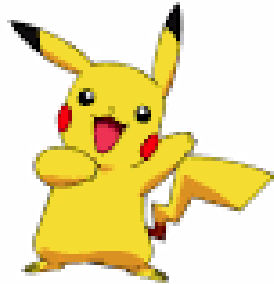
```
        // ...
```

```
    }
```

```
}
```

Если классы
находятся в
одном пакете,
то import не
нужен.

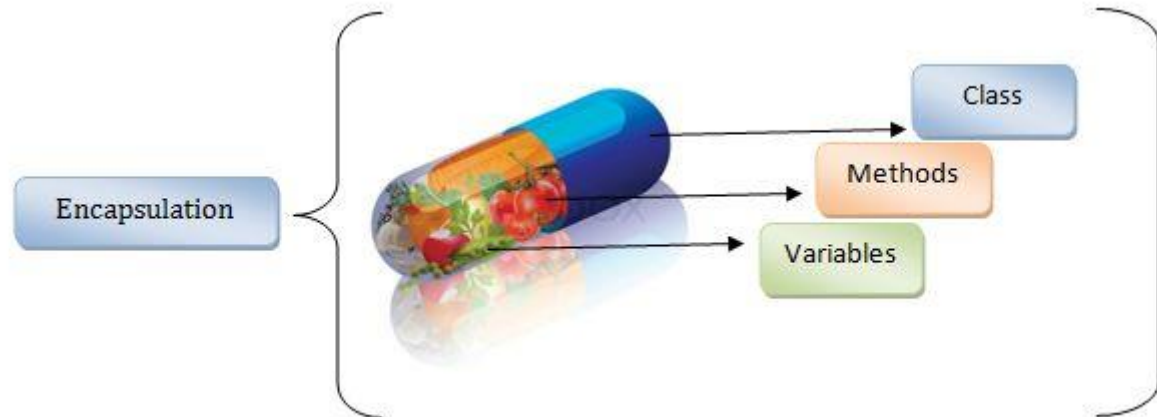
OOP principles



Pokemon
Name: Pikachu
Type: Electric
Health: 70
attack()
dodge()
evolve()

Fields

Methods



Inheritance

Поиск одинаковых атрибутов

- имя
- возраст
- рост
- вес
- пол

Поиск одинакового поведения

- сказать фразу
- применить навык
- присесть

Inheritance

```
package ru.ifmo.se.prog.examples;
```

```
public class Human {
```

```
    String name;
```

```
// имя
```

```
    boolean isMale;
```

```
    int age;
```

```
//
```

```
    возраст
```

```
    // ... остальные свойства
```

```
    public void sayPhrase() { }
```

```
    // ... остальные методы
```

```
}
```

Inheritance

```
package ru.ifmo.se.prog.examples;
```

```
public class Boy extends Human {
```

```
    public Boy() {  
        isMale = true;  
    }
```

```
        public void sayPhrase() {  
            // ...
```

```
        }  
    }
```

Inheritance

```
package ru.ifmo.se.prog.examples;
```

```
public class Girl extends Human {
```

```
    public Girl() {  
        isMale = false;  
    }
```

```
    public void sayPhrase() {  
        // ...  
    }
```

```
}
```


Inheritance

```
package ru.ifmo.se.prog;
```

```
import ru.ifmo.se.prog.examples.*;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Human[] array = new Human[2];
```

```
        array[0] = new Boy ();
```

```
        array[1] = new Girl ();
```

```
        for(Human human : array) {
```

```
            human.sayPhrase();
```

```
        }
```

```
    }
```

```
}
```

Reference type VS Object type

```
package ru.ifmo.se.prog;
```

```
import ru.ifmo.se.prog.examples.*;
```

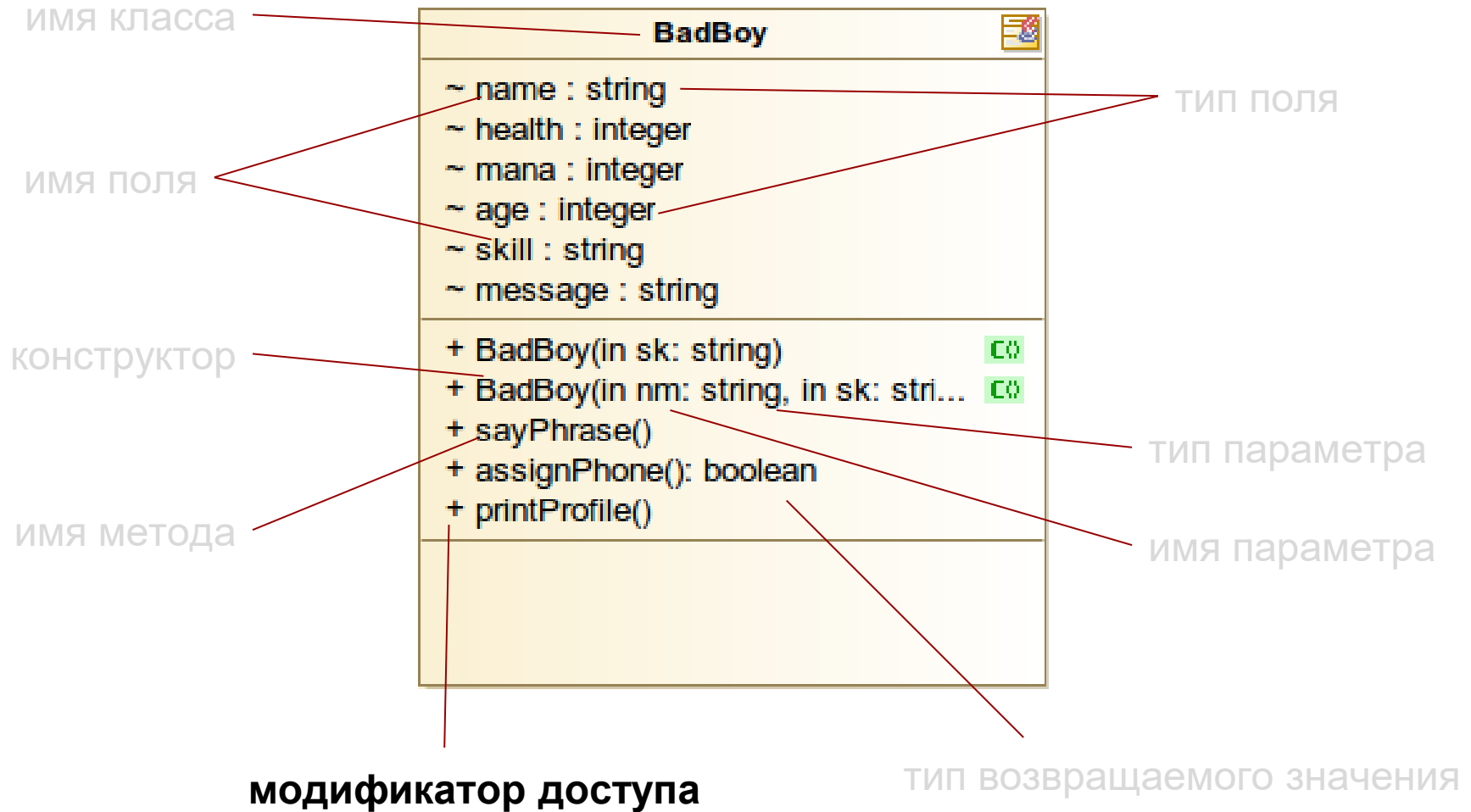
```
public class Main {
```

```
    public static void main(String[] args) {  
        Human human = new Boy();  
    }  
}
```

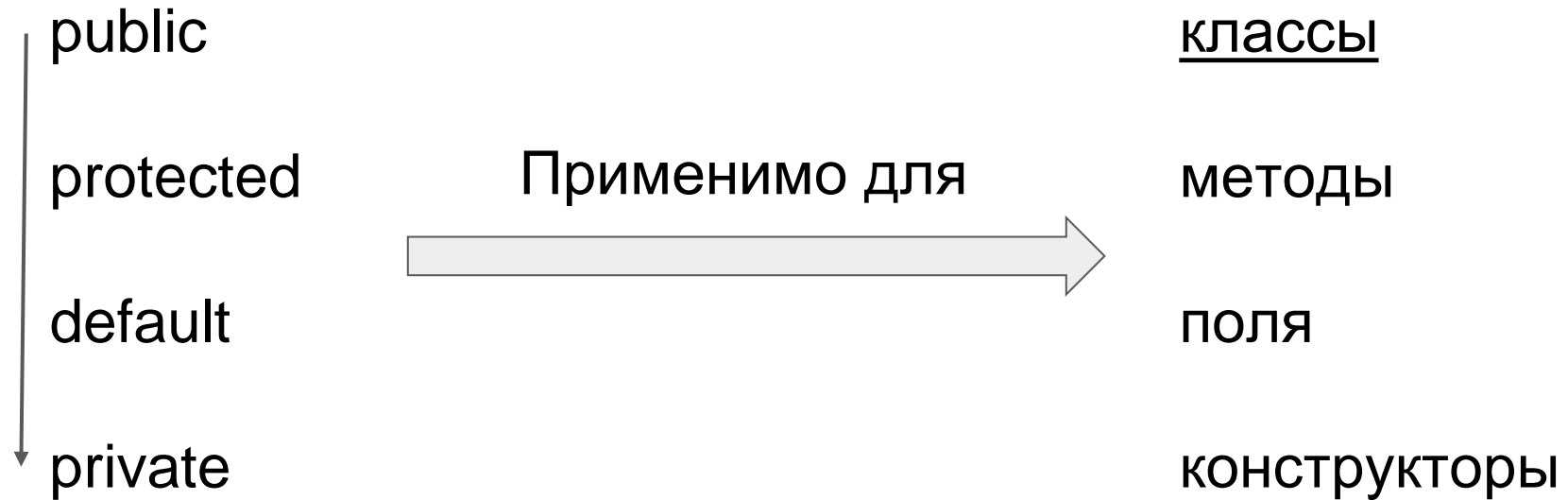
ТИП ССЫЛКИ

ТИП ОБЪЕКТА

Class description in UML



Access modifiers



How to name the file?

```
package ru.ifmo.se.prog.examples;
```

```
class Boy {
```

```
    private int age;
```

```
    int mana;
```

```
    protected int health;
```

```
    public int height;
```

// аналогично к методам и конструкторам. отсюда
вопрос:

```
        private Boy() {}  
    }
```

WAT?? Пока можно просто подумать...

How to name the file?

```
package ru.ifmo.se.prog.examples;
```

```
class Boy {
```

```
    // все поля и методы класса
```

```
}
```

```
class Girl {
```

```
    // все поля и методы класса
```

```
}
```

How to name the file?

```
package ru.ifmo.se.prog.examples;
```

```
public class Boy {
```

```
// все поля и методы класса
```

```
}
```

Boy.java

```
class Girl {
```

```
// все поля и методы класса
```

```
}
```

‘static’

```
public class Guy {
```

```
    public static Guy createClone(Guy guy) {  
        Guy clone = new Guy();  
        clone.name = guy.name;  
        clone.age = guy.age;  
        return clone;  
    }
```

```
// где-то в main методе
```

```
Guy myGuy = new Guy();  
Guy cloneOfMyGuy = Guy.createClone(myGuy);  
}
```


Why we wrote *'static'*

```
public class Main {  
    public static void main(String args[]) {  
    }  
}
```

JVM:
Main app = new Main();
app.main(args)

Why we wrote *'static'*

```
public class Main {
```

```
    public Main(int something) { ... }
```

```
    public static void main(String args[]) {  
    }  
}
```

JVM:

```
Main app = new Main(); // Error
```

How to name the file?

```
package ru.ifmo.se.prog.examples;
```

```
class Boy {
```

```
    private Boy() {}
```

WAT??



```
    private static Boy uniqueBoy;
```

```
    public static Boy getBoy() {
```

```
        if (uniqueBoy == null) uniqueBoy = new Boy();
```

```
        return uniqueBoy;
```

```
    }
```

```
}
```

‘this’ and ‘super’

```
package ru.ifmo.se.prog.examples;
```

```
public class Human {
```

```
    String name;                                // имя
```

```
    boolean isMale;                             // пол. true если муж.
```

```
    int age;                                    //
```

возраст

```
    // ... остальные свойства
```

```
    public Human() {
```

```
        // сначала бы вызвать родительский
```

конструктор...

```
    }
```

```
        public void applySkill(int age) { }
```

```
    // ... остальные методы
```

‘this’ and ‘super’

```
package ru.ifmo.se.prog.examples;
```

```
public class Human {
```

```
    String name;                                // имя
```

```
    boolean isMale;                             // пол. true если муж.
```

```
    int age;                                     //
```

```
    возраст
```

```
    // ... остальные свойства
```

```
    public Human() {
```

```
        // сначала бы вызвать родительский
```

```
конструктор...
```

```
    }
```

```
        public void applySkill(int age) { this.age = age; }
```

‘this’ and ‘super’

```
package ru.ifmo.se.prog.examples;
```

```
public class Human {
```

```
    String name;                                // имя
```

```
    boolean isMale;                            // пол. true если муж.
```

```
    int age;                                    //
```

```
    возраст
```

```
    // ... остальные свойства
```

```
    public Human() {}
```

```
    public Human() {
```

```
        this();
```

```
    }
```

```
    public void applySkill(int age) { this.age = age; }
```

‘this’ and ‘super’

```
package ru.ifmo.se.prog.examples;
```

```
public class BadBoy extends Human {
```

```
    public BadBoy() { super("Name"); }
```

```
}
```