Особенности программирования JAVA

Модуль 1. Основы языка Java

Информационное обеспечение курса

SAMSUNG IT АКАДЕМИЯ

Учебник Академии	https://myitacademy.ru/edu
Официальная документация по разработке	developer.android.com
Учебник для начинающих и продвинутых	https://startandroid.ru/
Сайт Александра Климова	developer.alexanderklimov.ru
Сборник ресурсов для Android-разработчиков	https://www.fandroid.info/tutorial-po-osnovam-yazyka-programmirovaniya-java-dlya-nachinayushhih/

Структура дисциплины

SAMSUNG IT АКАДЕМИЯ

* 5 модулей
* 8 лабораторных работ
* 2 практические работы
* 1 курсовая работа
* дифзачёт

№ п.п.	Контрольная точка	Оценка в баллах	Неделя
1	Контрольное тестирование по модулю 1	10	22
2	Лабораторная работа №1 Исследование жизненного цикла Activity	5	22
3	Лабораторная работа №2 Создание однооконного интерфейса Android-приложения	5	23
4	Лабораторная работа №3 Многооконные приложения и фрагменты	5	23
5	Контрольное тестирование по модулю 2	10	24
6	Лабораторная работа №4 Работа приложения с файловой системой устройства	5	24
7	Лабораторная работа №5 Графика в Android	5	25
8	Контрольное тестирование по модулю 3	10	26
9	Лабораторная работа №6 Использование коллекций в Android-приложениях	5	26
10	Практическая работа №1	5	27
11	Лабораторная работа №7 Использование локальной базы данных в мобильном приложении	5	27
12	Контрольное тестирование по модулю 4	10	28
13	Практическая работа №2	5	28
14	Лабораторная работа №8 Retrofit. Rest-взаимодействие	5	29
15	Контрольное тестирование по модулю 5	10	30

Технология Java

SAMSUNG IT АКАДЕМИЯ

«Под программным обеспечением Java или средой выполнения Java подразумеваются также Java Runtime, среда выполнения, модуль выполнения, JRE, виртуальная машина Java, виртуальная машина, Java VM, JVM, VM, подключаемый модуль Java, дополнение Java и загружаемые файлы Java.»

ORACLE

Объектно-ориентированная структура программы

Стандарт Си

Автоматическая расстановка указателей

Замена goto на break с меткой

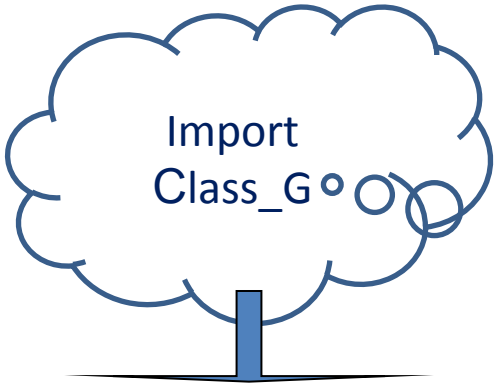
Тернарная операция – именно ОПЕРАЦИЯ

Защищённые классы видны на уровне пакета

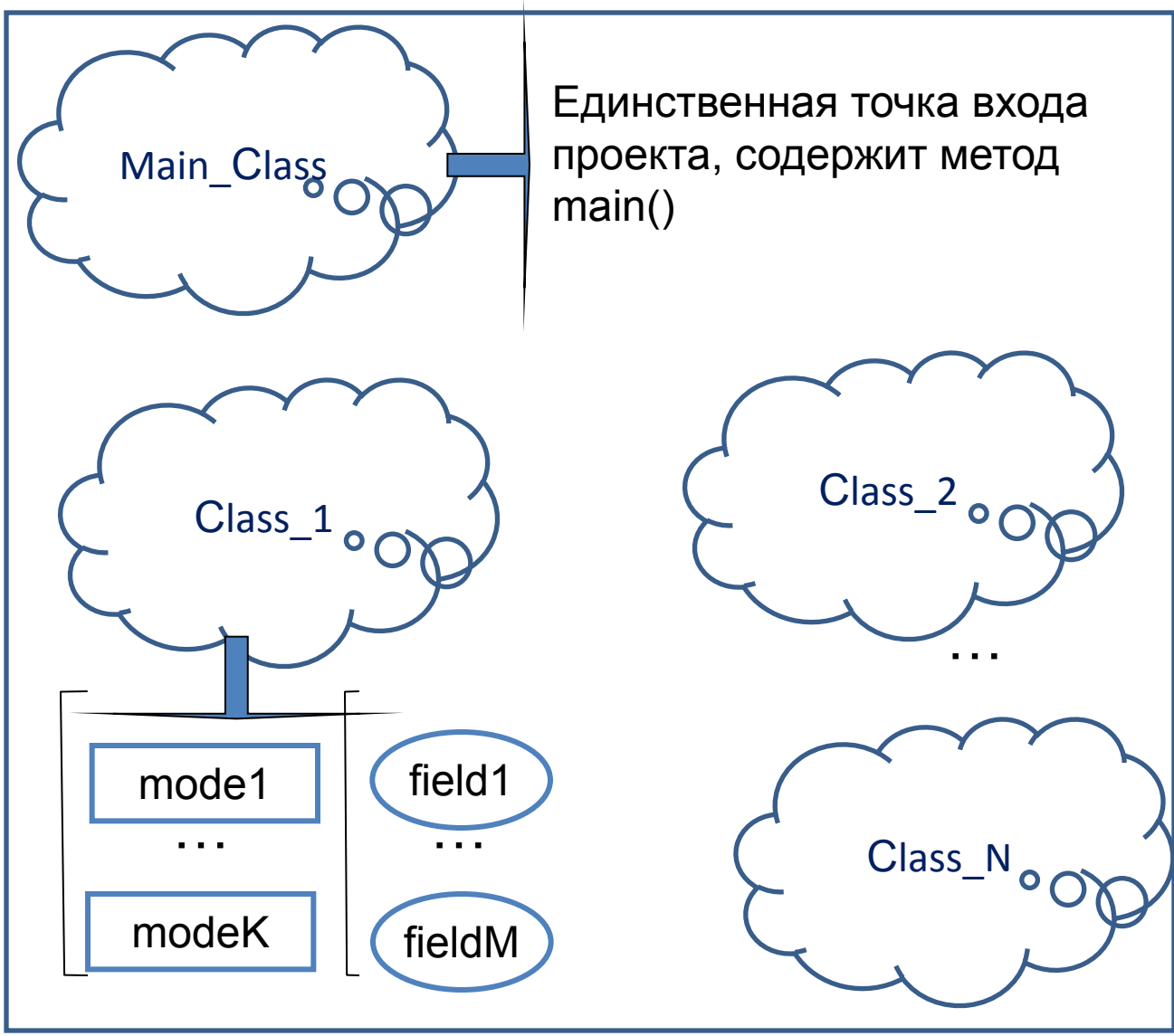
Для сборки мусора неиспользуемые объекты следует закрывать!

Java-проект

Размещается в пакетах (папках)



Классы из других пакетов классов, используемые в текущем пакете



Типы данных Java

SAMSUNG IT АКАДЕМИЯ

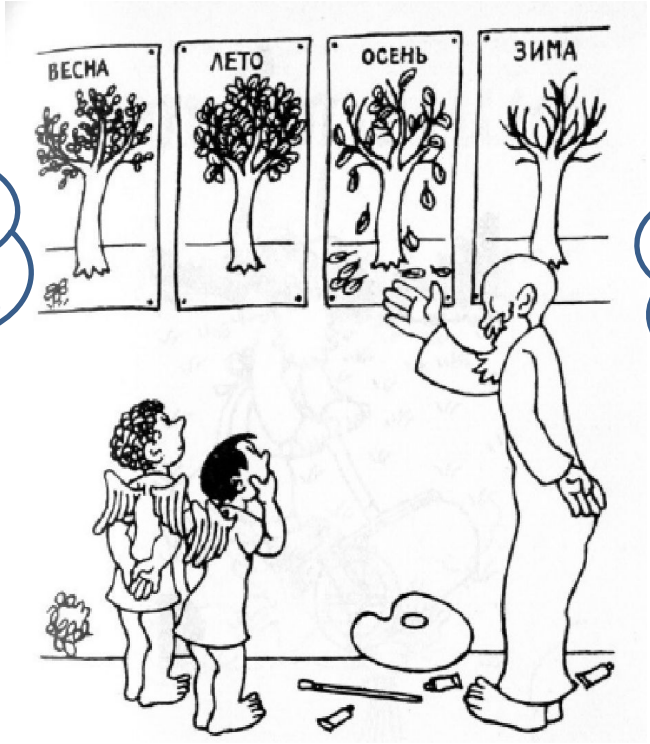
Базовые (Primitive)

целочисленные

вещественные

знаковый

логический



Объектные (Reference)

массивы

классы

интерфейсы

Классы JAVA

- Имя класса совпадает с именем файла, в котором он размещается
- В одном файле могут описываться несколько классов, но только один из них **публичный** и **его имя** - имя файла
- Классы могут быть вложены в другие классы (агрегированы)
- Классы могут быть анонимными (класс-объект без имени)
- Статические классы сами выступают в роли своего объекта
- Из статического метода можно вызвать только

Структура класса

SAMSUNG IT АКАДЕМИЯ

```
access_modifier  class ClassName {
fields & modes
}
```

```
class Sample {
 private int ifield;
        public int getIfield() {
        return ifield;}
    public void setIfield(int ifield) {
        this.ifield = ifield;}
    public Sample() {}
    public Sample(int ifield) {
        this.ifield = ifield;}
    int incIfield(int diff){
            return ifield + diff;}
    double divIfield(int denom){
  return (denom == 0)? 0 : (double)ifield/denom;}
}
```

Унарные арифметические операции

-	-x	унарный минус	изменяет знак выражения на противоположный
+	+x	унарный плюс	ничего не производит
~	~x	побитовое дополнение (int)	инвертирует биты в коде числа
++	++x, x++	инкремент (int)	увеличивает число на 1
--	--x, x--	декремент (int)	уменьшает число на 1

Бинарные арифметические операции

-	x-y	вычитание	вычитает из значения левого операнда значение правого
+	x+y	сложение	складывает значения операндов
*	x*y	умножение	умножает значения операндов
/	x/y	деление	делит значение левого операнда на значение правого
%	x%y	остаток	возвращает остаток от деления значения левого операнда на значение правого

Бинарные побитовые операции

SAMSUNG IT АКАДЕМИЯ

&	x&y	побитовое «И» (конъюнкция)	выполняет поразрядную конъюнкцию кодов аргументов
|	x|y	побитовое «ИЛИ» (дизъюнкция)	выполняет поразрядную дизъюнкцию кодов аргументов
^	x^y	побитовое «исключающее ИЛИ» (XOR)	выполняет поразрядную неравнозначность кодов аргументов
<<	x<<y	левый сдвиг	перемещает биты кода аргумента x на y позиций влево
>>	x>>y	правый сдвиг с учетом знака	перемещает биты кода аргумента x на y позиций вправо (бит знака смещается)
>>>	x>>>y	правый сдвиг без учета знака	перемещает биты кода аргумента x на y позиций вправо (бит знака НЕ смещается)

Сравнение и тернарная операция

SAMSUNG IT АКАДЕМИЯ

У объектов сравниваются указатели!

==	x==y	равно	возвращает true, если **числовые** значения параметров равны.
!=	x!=y	не равно	возвращает true, если числовые значения параметров не равны
> >=	x>y x>=y	больше больше или равно	возвращает true, если числовое значение параметра x больше (или равно) значения параметра y
< <=	x<y x<=y	меньше меньше или равно	возвращает true, если числовое значение параметра x меньше (или равно) значения параметра y

<условие> ? <значение_если_true> : <значение_если_false>

возвращает в обоих случаях один и тот же тип данных!!!

Класс String

SAMSUNG IT АКАДЕМИЯ

Объект входного и выходного потока данных

```
String hello1 = "Здравствуйте!";
String hello2 = new String("Здравствуйте!")
```

Тип	Метод	Описание
String	+	конкатенация
boolean	equals(String obj)	эквивалентность
int	length()	длина
char	charAt(int index)	Символ на позиции index
int	compareTo(String anotherString)	Сравнение по буквам с учетом

Итератор по массиву

SAMSUNG IT АКАДЕМИЯ

```
for(type param: array) {
work(param);
  }
```

```
int[] iArray = new int[10];
    for (int p:iArray) {
        System.out.print(p+"\t");
    }
 String sArray[] = new String[3], s="";
    for (String word: sArray) {s = s + word + " ";
    }
```

Прерывание вложенных циклов

SAMSUNG IT АКАДЕМИЯ

```
label:
  out_circle {…
        in_circle { …
                    break label;
                    …
        }
  }
```

```
 met:
for(int f = 0; f < F; f++){
int k =
    (int)(Math.random()*f)+1;
while(k < K){
  System.out.print(10*f+k);
  if(f - ++k < 5) break met;
    }
}
```

Статические классы для базовых типов

Примитивный тип	Соответствующий класс-оболочка
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

```
String great = "5";
int  excellent = Integer.parseInt(great);
```