Android разработка. Подготовительный курс

Лекция 3. Базовый синтаксис: области видимости. Методы. Классы

Лекция 3. Повторение - мать учения

- Вывод в консоль осуществляется командой:
  System.out.println();
  где в скобочках нужно указать что мы хотим напечатать
- Ввод с консоли осуществляется с помощью класса Scanner. Нужно создать объект класса, настроить его на поток ввода с консоли и не забыть закрыть после окончания работы.
- Чтение цифр осуществляется:
  sc.nextInt();
  который возвращает в программу цифру, введенную в консоли.
- Чтение строк осуществляется:
  sc.next();
  который возвращает в программу строку, введенную в консоли.

Лекция 3. Повторение – мать учения

- Для того чтобы объявить переменную необходимо написать
  int a;
  , где int – тип переменной, a – название. Этой командой мы резервируем имя a за переменной типа int.
- Имя переменной может быть любое, главное чтобы не повторялось внутри одной области видимости.
- Переменная может быть как типа так и класса.
- Для того чтобы внести в переменную значения то нужно написать имя переменнной и после равно и значение:
  a=19;
  Это можно сделать как при объявлении так и позже. Процесс установки значения называется инициализация.

Лекция 3. Повторение – мать учения

- Условные операторы if else проверяют условие которое им необходимо передать.
```
if(a>3){
//если true
} else {
// если false
}
```
- Блок else может отсутствовать.
- В условие можно передавать переменную типа boolean.

Лекция 3. Повторение – мать учения

- Циклы бывают 3-х видов: со счетчиком, с пред условием, с пост условием.
- Цикл со счетчиком:
```
for(int i=0; i<10; i++){
}
```
- Цикл с пред условием:
```
while(a>3){
}
```
- Циел с пост условием:
```
do{
} while(a>3);
```

Урок 3. Базовый синтаксис. Области видимости

- Области видимости обозначаются фигурными скобками: {//что-то внутри}
- Внутри фигурных скобок пишется один целостный блок работы(алгоритм действий)
- Внутри фигурных скобок не видно переменные вне фигурных скобок
- Каждый очередной этап вложенности выделяется табуляцией. Например:

```
{
        //Первый уровень вложенности
        {
                //Второй уровень вложенности
        }
}
```

- Каждая новая строчка должна заканчиваться ; если это не блок области видимости

Урок 3. Метод

- Метод это блок, который предназначен для отделения своей зоны видимости.
- Метод описывается следующим образом:
```
public void printHello(){//объявление метода и старт зоны метода
        System.out.println("Hello, world!");//тело метода
}//объявление окончания метода
```
- Метод можно вызвать из любого места, из которого его видно.
- Метод служит для логического отделения блока действий, описанного в нем. Метод с примера напишет Hellom world! В консоль.

Урок 3. Метод

- Рассмотрим подробнее описание метода:
  public void printHello()
- Первая часть public - это описание того где будет виден этот метод. Пока не придавайте значения, об этом в дальшейших уроках. Отмечу только то что public – это везде.
- Вторая часть void – это описание возвращаемого значения. void означает «ничего» т.е. значение не возвращается. Кроме void там может писаться любой int или String или любой другой тип/класс.
- Например когда мы вызывали scanner.next(); он возвращал String который надо было присвоить в переменную или использовать другим путем.

Урок 3. Классы

- Классы – это отдельные блоки которые являют собой отображение логической сущности(объекта). Он нужен для отделения логики работы с одной логической моделью в себе. Проще объяснить на примере:
- Допустим, нам нужен класс Человек(Human). У него будет рост(height), возраст(age), имя(name).
- public class Human{
  public float height;
  public int age;
  public String name;
  }

Урок 3. Классы

- Для того чтобы его использовать необходимо создать его экземпляр. Ведь людей у нас может быть много. Давайте создадим Васю и Машу.
- public static void main(Stirng[] args){
  Human vasya = new Human();
  Human masha = new Human();
  }
- Объекты(экземпляры) класса создаются ключевым словом new.
- Однако пока мы только назвали переменные, но не дали имя внутри объекта.

Урок 3. Классы

- Для того чтобы дать имя нам необходимо дать значение переменной name которую мы уже объявили внутри класса. Для этого мы будем использовать специальный метода, называемый конструктор.
- public class Human{

```
        …
        public String name;

        public Human(String name){
                this.name = name;
        }
}
```

- И

```
public static void main(Stirng[] args){
        Human vasya = new Human(“Вася”);
        Human masha = new Human(“Маша”);
}
```

Урок 3. Классы

- Теперь добавим один метод который будет означать рост нашего человека. Но прежде нужно инициализировать еще 2 переменных.
- public class Human{
```
        public float height;
        public int age;
        public String name;

        public Human(String name){
            this.name = name;
            age = 1;
            height = 50f;
        }
}
```
- По сути то что мы передаем в скобках это то что мы можем использовать внутри метода. Нам не обязательно что-то нужно.
- Обратите внимание, перед name стоит this, а перед age и height – нет. this означает что эта переменная берется из области видимости на 1 уровень выше.

Урок 3. Классы

- Ну и опишем собственно наш метод рости(grow).
- public class Human{

```
public float height;
public int age;
…

public void grow(){
    height+=12;
    age++;
}
}
```

Урок 3. Классы

- Добавим немного логики к методу grow()
- public void grow(){

```
        if(age<25)
            height+=12;
        age++;
        if(age==100)
            System.out.println("Поздравляю, "+ name + " - долгожитель!");
    }
```

Урок 3. Классы

- Вырастим нашего Ваню до 100, а Машу до 99 и увидим кто из них долгожитель =)
- public static void main(Stirng[] args){

```
        Human vasya = new Human(“Вася”);
        Human masha = new Human(“Маша”);
        for(int i = 0; i<100; i++){
            vasya.grow();
        }
        for(int i = 0; i<99; i++){
            masha.grow();
        }
}
```

Урок 3. Класс + дз

- Реализовать класс Человек(Human) из урока у себя в проекте.
- Переописать логику таким образом чтобы до 25 росли на 12, а после 60 уменьшались на 2.
- Добавить логику чтоб после 150 была смерть и рост останавливался.
- Вырастить Олега и Дашу на 120 и 200 соответственно.
- Сделать возможность сменить имя.

Урок 3. Класс + дз. Вам не выжить

- Сделать класс Машина(Car).
- У него должны быть 3 поля: модель(String), кол-во литров бензина в км (int), кол-во бензина в баке(float).
- Сделать метод заправки бензина в машину.
- Сделать метод проезда n км. В нем описать следующую логику:
  Если бензина хватает то проехать и написать в чат что поехали, остаток столько то бензина.
  Если бензина не хватает то проехать сколько хватит и написать сколько проехали и что бензин закончился.

Урок 3. Класс + дз. Немного алгоритмов

- Палиндром – число которое пишется и читается одинаково как слева направо так и справа налево.
- Написать программу которая будет выяснять является ли число палиндромом.

Куда отправлять ДЗ?

db@bigdig.com.ua
Обязательно подписывайте.

Файлики *.java запихиваете в архив и присылаете по почте.

Файлики лежат по пути место_хранения_проекта\untitled1\src\com\company\
(где untitled1 – название проекта, com\company – название пакета)