HELLO WORLD

package com.company; paketi mogyt o6ediniat6 klasi tobest6 organizovat6 v pronstranstve imen   
import java.util.Scanner;

public class Main { OOP eto klasi I ob6ekti eto osnovnoi klass kotorii mi delaem pablik

privet class Main { bydet dostypen iskluchitel6no v etom pakete

public static void main(String[] args) {

//public dostypen po vsudy static void fynciia nichego ne vozrachaet main nazvanie fyncii (String[] args) I prinimaet massiv s nazvaniem ARGSargyment { eto fynkcii main vipolniaet6sa pri starte programi

; oznachaet chto mi zakrivaem stroky posle kazdoi strochki krome kogda stavim {

//veto komentarii dlia koda

System.*out*.print("Hello world !");

//System ssilaemca k system .*out ssilaemca k fynkciam ob6ekta* .print eto bydet idti vse v odnoi stroke or println chto oznachaet s novoi stroki("Hello world !");

PEREMENNIE

**CELIE CHISLA**

* **byte** - предназначен для хранения целых чисел. Диапазон чисел от -128 до 127, если число будет большим 127 или меньшим -128 , то выдастся ошибка;
* **short** - предназначен для хранения целых чисел. Диапазон чисел от -32768 до 32767;
* **int** - предназначен для хранения целых чисел. Диапазон чисел от -2147483648 до 2147483647;
* **long** - предназначен для хранения целых чисел. Диапазон чисел от –9 223 372 036 854 775 808 до 9 223 372 036 854 775 807.

**ChISLA S TOCHKOI**

* **float** - для создания чисел с плавающей точкой. Диапазон чисел от от -3.4\*1038 до 3.4\*1038; obezatel6no v konce postavit6 f

float num\_1 = 23.45f;

* **double** - для создания чисел с плавающей точкой. Диапазон чисел от от ±4.9\*10-324 до ±1.8\*10308. obezatel6no v konce postavit6 d

double num\_2 = 1.8\*10.308d;

**Прочие типы данных**

* **boolean**- логический тип данных. Предназначен для записи истинного (true) или ложного (false) значения;

boolean isTrue = true; eto bydet 1

boolean isFalse = false; eto bydet 0

* **char** - тип данных для работы с символами. Позволяет поместить в одинарных кавычках какой-либо символ;

char sym = 'R'; pomewaem odin konkretnii simvol v odinarnih skobkah ‘ ‘

* **String** - тип данных для текста. В двойных кавычках можно сохранить одно или множество слов.

String str = "Hello world"; mozno pomestit6 celii tekst no nyzno postavit6 “ “

Inoga voznikaut owibki I vetih slychaiah nyzno propisat6 importirivat6 dopolnitel6nii paket klass

import java.lang.String;

public static void main(String[] args) {  
 byte num;  
 byte num = 123;

// byte tip dannih peremennoi num eto prosto nazvanie ; eto zakritie strochki

// byte num = 123 eto prisvoenie znachenia;

Single line comment // toy,boy…….

Multi line comments /\*toy,boy

Zoy,fort \*/

POLYCHENIE INFORMACII OT POL6ZOVATELIA

Dla nachala nam nebhodimo ispol6zovat6 takoi klass kak Scanner

1. nam nyzno zagrystit6 ego v nawem proekte

import java.util.Scanner;

teper6 mi mozem ispol6zovat6 scaner

package com.company;  
import java.util.Scanner;  
  
  
public class Main {  
  
 public static void main(String[] args) {  
 Scanner str eto imia mi piwem kak s peremenoi = new eto ob6ekt klasa Scanner(System.*in eto oznachaet chto mi prosto bydem prinimat6 informaciu*);  
 System.*out*.print(str.nextLine()); nyzno bilo propisat6 ne prosto str a str.nextLine()

Dopolnitel6no mi mozem naoisat6 stroky s tekstom

System.*out*.print( "Your string is:" + str.nextLine());

Kogda nyzno ymnozit6 chisla

package com.company;  
import java.util.Scanner;  
  
  
public class Main {  
  
 public static void main(String[] args) {  
 Scanner num = new Scanner(System.*in*);  
 int first, second, result; I sdelaem 3 peremennie  
 System.*out*.print("Enter first num:"); prosim pol6zovatelia chtobi on vvodil chisla  
 first = num.nextInt(); peremennaia first bydet zapisano chislo v kotoryu vvedet pol6zovatel6  
 System.*out*.print("Enter second num:");  
 second = num.nextInt();  
 result = first + second; v peremennoy result mi zatragivaem temy matematicheskih operacii  
 System.*out*.println("Result is - " + result);

esli mi hotim ves6 rezyl6tat podelit6 na 2

result = first + second;  
result /= 2;  
System.*out*.println("Result is - " + result);

information result++; eto mi dobavliaem +1 libo – eto mi otnimaem 1

Pri dilenii lychwe ispol6zovat6 tipi dannih float

* вычитание - z = y - x;
* сложение - z = y + x;
* умножение - z = y \* x;
* деление - z = y / x (на ноль делить нельзя);
* остаток при делении - z = y % x; esli 10 % 3 bydet = 1

YSLOVNIE OPERATORI IF or ELSE and ELSE IF

public static void main(String[] args) {  
 Scanner num = new Scanner(System.*in*);  
 int first;  
 System.*out*.print("Enter first num:");  
 first = num.nextInt();  
  
 if (first == 10) vvodim yslovniy operator if chto oznachaet esli I potom stavim nekoe yslovie (first == 10)  
 System.*out*.print("Num is 10");

esli y nas neskol6kp strochek koda v yslovnom to mi obezatel6no dozni zakrivat6 skobkami { }

if (first == 10) {  
 System.*out*.print("Num is 10");  
 System.*out*.print("\n");  
}

mi mozem eto proveriat6 ne tol6ko na ravenstvo

if (first > 10) { if (first < 10) { if (first <= 10) { if (first >= 10) {

vmesto chisla mozem postavit6 drygyu peremennyu

int first, second = 50;  
boolean isTrue = false;  
System.*out*.print("Enter first num:");  
first = num.nextInt();  
  
isTrue = first == 51;  
  
if (first >= second && isTrue) {

else,

else if

} else if (first == 45) eto iavliaet6sa dopolnitel6nim ysloviem  
 System.*out*.print("Num is equel 45");  
else { proverachnii metod esli pervaia konstrykciia ne srabativaet  
 System.*out*.print("Num is lower than " + second);  
}

Esli mi hotim sdelat6 ewe odny proverky na ewe odno yslovie

Scanner num = new Scanner(System.*in*);  
int first, second = 50;  
boolean isTrue = false; dlia etogo sozdaem peremennyu BOOLEAN  
System.*out*.print("Enter first num:");  
first = num.nextInt();  
  
isTrue = first == 51;  
  
if (first >= second && isTrue) { mi dopisivaen dva && kotorie oznachaut И esli eti 2 yslovia bydyt True togda vipolnit6sa etot kod  
 System.*out*.print("Num is 10");  
 System.*out*.print("\n");  
} else if (first == 45)  
 System.*out*.print("Num is equel 45");  
else {  
 System.*out*.print("Num is lower than " + second);

if (a != b && a > b) {

// Код будет выполнен, если и первое, и второе условие

// окажутся верными

}

if (a < b || a == b) {

// Код будет выполнен, если или первое, или второе условие

// окажется верным

MOZNO VSE NAPISAT6 BOLEE KRATKO CHEM ETO PREDSTAVLENO SVERHY

isTrue = first == 51 ? togda mi stavim true : inache mi stavim false;

eto vigliadit dovol6no prostin6ko no ne standartno

ili kogda est6 BOOLEON mozno napisat6 I tak isTrue = first ==51;

KONSTRUKCIA YSLOVNII OPERATOR

switch () { proveriem peremennyu na konkretniy sluchai

switch (first) { propisivaem first peremennyu  
 case 51: propisivaem case I proveriam peremennyu na konkretnii slychai.  
 System.*out*.print("Num is equel 51"); esli ona bydet ravna 51 to viidet takoe soobwenie  
 case 48:  
 System.*out*.print("Num is equel 48"); keisov mozno sdelat6 skol6ko ugodno   
 break; eto nebhodimii operator kotorii govorit chto mi zakonchili vipolnuat6 kod  
 default: :est6 znachenie po ymolchaniu esli niodin case ne vipolnin to bydet znachenie po ymolchaniu  
 System.*out*.print("Num is equal something");  
} posle default break stavit6 ne nyzno

CIKLI V IAZIKE JAVA blok koda kotoriy mi vipolniaem neskol6ko raz

EST6 “3” cikla

1. for (int dlia obiavlenie peremennoi “int” mi ee ykazivaem priamo tyt i = 0; i < dolza vipolniat6tsa do 10; i++ dolzna yvelichivat6sa) {

public static void main(String[] args) {  
 for (int i = 1; i < 11; i++) { posle i-=10 to bydet otnimat6 po 10  
 System.*out*.println(i);

1. while v eton ciklle vse naoborot int vinosit6sa otdel6no I ymnozenie toze vivodit6sa otdel6no” J += 2”

int j = 0;  
while (j <= 20) {  
 System.*out*.println(j);  
 j += 2;

osobenosti while mi mozem proveriat6 boalean peremennie pisat6 TRUE

boolean isTrue = true;  
int j = 0;  
while (isTrue) {  
 System.*out*.println(j);  
 j += 2;

no etot cikle bydet bezkonechnim, ego nyzno zaverwit6. Esli mi dopiwem yslovnii operator if

boolean isTrue = true;  
int j = 0;  
while (isTrue) {  
 System.*out*.println(j);  
 j += 2;  
 if (j > 10)  
 isTrue = false;

1. do while cikle proveriaet6sa til6ko odin raz on vipolniet6sia a potom provereriaet6sa yslovie esli yslovie ne verno ono zakanchivaet cikle

int x = 0;  
do {  
 x++;  
 System.*out*.println(x);  
} while (x < 0);

OPERATORI

Breake ochen6 ydobno prosto viti iz cickla

boolean isTrue = true;  
int j = 0;  
while (isTrue) {  
 System.*out*.println(j);  
 j += 2;  
 if (j > 10)  
 break;

Continue on propyskaet 1 operaciu

boolean isTrue = true;  
int j = 0;  
while (j < 100) {  
 j += 1;  
 if (j % 2 == 0)  
 continue;  
 System.*out*.println(j);

Return on ispol6zyetsa bol6we v metodah I funciah hotia ochen6 pohoz na break

MASSIVI

Nekoe peremennoe v kotoryu mi mozem zakluchit6 neskol6ko znachenii

int name;

na danii moment eto prosto peremennaia

dlia togo ctobi sdelat6 Massiv nyzno postavit6 [ ]

int[] namen = new int [5];

v massive vse indeksi nachinaut6sia s 0

int[] namen = new int [5];  
name[0] = 23;  
name[2] = 12;

no mozno poiti I bolee prostim pytem

int[] name = new int [] {23,34,56};

dlia togo chtobi vivesti na ekran mi dolzni ykazat6 index( index 1 eto bydet 34)

System.*out*.println(name[1]);

Probnaia programa

package com.company;  
import java.util.Scanner; 1. Podkluchae klas scanner pozvoliet schitivat6 danii kotorii vvell pol6zovatel6  
  
public class Main {  
  
 public static void main(String[] args) {  
 int [] a; sozdaem massiv no ne videliaem pamiat6 pod nego tak kak ne znaem skol6ko pol6zovatel6 ix vvedet  
 int n; prosto peremennaia  
 Scanner in = new Scanner (System.*in*); sodaem ob6ekt in tipa clasa Scanner kotorii bydet schitivat6 danie System ot pol6zovatelia in  
 System.*out*.print("Enter numbers of array: "); tyt mi prosto vivodim informaciu na ekran  
 n = in.nextInt(); pol6zovatel6 vvodit v in.nextInt I eto vse idet v peremennuy n  
 a = new int[n]; blagodaria peremennoi n videliaem pamiat6 pod masiv a

teper6 idet obichniy cikle  
 for (int i = 0; i < n; i++) { ot nachalo 0 s i indeksom 0 I do < elementa kotorii vvel klient n I dobavlenie po 1 ++  
 System.*out*.print("Enter a [" + i +"] ="); tyt mi sprashivaem pol6zovatela chtobi on vvel konkretniy element v massiv  
 a [i] = in.nextInt(); I dal6we mi eto vvodim konketnii element massiva a s indeksom i vvodim danie s klaviatyrii in.nextInt();  
 }

package com.company;  
import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Scanner at = new Scanner (System.*in*);  
 System.*out*.print("Enter numbers of array: ");  
 int [] a;  
 int n;  
 n = at.nextInt();  
 a = new int[n];  
 for (int i = 1; i < n; i++) {  
 System.*out*.print("Enter an [" + i +"] =");  
 a [i] = at.nextInt();  
 }  
 }  
  
}

MNOGOMERNIE MASSIVI

Eto obichnii odnomernii masiv

float[] array = new float[] {12.22f,23.33f}; v kotorih est6 2 elementa

tak vigliadit obichnii dvymerniy masiv

float[] array = new float[] {{12.22f,23.33f},{23.23f,23.33f}};

voobwe dvomernii (mnogomernii) masiv zapisivat6 proche I ydobnee tak

char[][] array = new char[3][3]; mi stavim 2 kvadratnie [][] I posle dolzi ykazat6 dliny kazdogo massina [3][3];  
array[0][0] = 'C'; s ideksom 0 I posle etogo pishem simvol C  
System.*out*.print(array[0][0]);

esli vdryg voznikaet owibka (eto vozmozno v starih versiiah)

nyzno pod kazdii otdel6nii masiv videlat6 otdel6no pamiat6

for (int i = 0; i < array.length; i++) LENGTH eto dlina v massive   
 array[i] = new char [3]; v etoi stroke m videliaem otdel6no dliny pod kazdii massiv

FYNCII v JAVA

Eto podprogrami na kotrie mi mozem ssilat6sa

package com.company;  
import java.util.Scanner;  
  
public class Main {  
  
 public eto modifikator dostypa static void eto tip vozrochaemih znachenii v donon sluchae eto oznachaet chto mi nichego ne vozrachaem etoi fyncii > main eto fynkciia(String[] obichnii massiv args eto strokovii masiv ) {

public static void main(String[] args) {  
 int temp = *func* (3,4); ssilka k fyncii I polychenie nekogo chisla  
 System.*out*.print(temp); eto fynkcia vivodit vse na ekran  
  
}  
public static mi obezatel6no dolzni ykazat6 modikator dostypa int tip dannih kotorii mi peredaem po okonchanii etoi funcii func nazvanie funkcii (int a eto prosto chislo, int b eto stepen6) v kryglih skobkah peredaut6sa parametri {eti skobki telo funkcii  
 int result = 1; prosto dabavili odnu peremenyu  
 for (int i = 0; i < b; i++) vozvishenie v stepeni mi dolzni delat6 vcikle  
 result \*= a; peremennuy result miumnozaem na A  
 return result; nyzno vkluchit6 chtoto dlia vozrata tak kak mi ispol6zovali int

public static void main(String[] args) {  
 int temp = *func* (3,4);  
 System.*out*.print(temp);  
  
}  
public static int func (int a, int b) {  
 int result = 1;  
 for (int i = 0; i < b; i++)  
 result \*= a;  
 return result;

esli postavit6 VOID ona bi nichego ne vozrachala

public static void main(String[] args) {  
 *func* (3,4);  
  
  
}  
public static void func (int a, int b) {  
 int result = 1;  
 for (int i = 0; i < b; i++)  
 result \*= a;  
 System.*out*.print(result);

primer

package com.company;  
import java.util.Scanner;  
  
  
public class Main {  
  
 public static void main(String[] args) {  
 *gog* (3,4);  
 *readArray*();  
 }  
 public static int[] readArray () {  
 int[] r;  
 int e;  
 Scanner in = new Scanner(System.*in*);  
 e = in.nextInt();  
 r = new int [e];  
 for (int i = 0; i < e; i++)  
 r[i] = in.nextInt();  
 return r;  
 }  
 public static void gog (int a, int b) {  
 int ressult = 0;  
 for (int z = 0; z < b; z++)  
 ressult += a;  
 System.*out*.print(ressult);  
 }  
}

OOP

**Класс** в случае с роботом – это его чертёж. Экземпляром класса (объектом) называет целый робот, который создан точно по чертежу.

**Наследование** – это добавление полезных опций к чертежу робота. К примеру, берем стандартный чертёж робота и дорисуем к нему лазеры, крылья и броню. Все эти дорисовки мы сделаем в классе наследнике, основной функционал которого взят из родительского класса.

**Полиморфизм** – это общий функционал для всех роботов и не важно что каждый робот может очень сильно отличаться друг от друга. К примеру, в главном классе мы указываем возможность передвижения для всех последующих роботов. Далее в классе наследнике мы можем дополнительно указать возможность левитации для робота, в другом же классе укажем возможность передвижения по воде и так далее. Получается, что есть общий функционал что записан в главном чертеже, но его можно переписать для каждого последующего робота (для каждого наследника).

А **инкапсуляция** является для нас бронёй, защищающей робота. Под пластырем брони находятся уязвимые элементы, вроде проводов и микросхем. После прикрытия брешей с помощью брони (protected или private), робот полностью защищён от внешних вмешательств. По сути, мы делаем доступ ко всем полям лишь за счёт методов, тем самым

SOZANIE KLASSA A TAKZE OBEKTOV

V Java novie klassi priniato sozdavat6 v novih failah. Nazvaniai klassov priniato nazivat6 s bol6shih bykv

Существует три модификатора доступа:

* public - данные будут видны повсюду, как в классе, так и вне его;
* protected - данные будут видны только в классе, где они были созданы, а также в классах наследниках;
* private - данные будут видны только в классе, где они были созданы.

Sozdali takoi klass kak cat

package com.company;  
  
public on imeet modificator dostyp pablic class Cat {  
 public nyzno dopisivat6 modifikator dostypa int peremennaia height = 180; peremennie v klase priniato nazivat6 poliami Fynkcii v klase priniato nazivat6 metodami

public void say (String name) {  
 System.*out*.print("Hello, " + name );  
  
 }

eti klassi pomogaut sozdavat6 dlia vseh personazei odinakovyu funkciu

package com.company;  
  
public class Cat {  
 public int height = 180;  
  
 public void say (String name) {  
 System.*out*.print("Hello, " + name );  
  
 }

package com.company;  
import java.util.Scanner;  
  
  
public class Main {  
  
 public static void main(String[] args) {  
 Cat teper6 mi sozdaem ob6ekt klassa CAT victor eto ego imia = new Cat();I videlili pod nego pamiat6  
 victor. Kogda stavim tochky mi dostykivaemsia do vseh polei I metodov height = 120; eto mi personal6no vivodim razmer pod etot personal   
 victor.say(" rubi ");  
 System.*out*.println(victor.height);  
 Cat myrzik = new Cat();  
 myrzik.say ( " yrodi ");  
 System.*out*.println(myrzik.height);

KONSTRYKTARI PRI SoZDANII KLASSA

Konstryctor prisydstvyet vo vseh klassah, mi ego ne videm tak kak on syoit po ymolchaniu. Konstryktori sozdaut ekzemliari klassov

Konsryktorov mozet bit6 neskol6ko v odnom klase mi mozem meniat6 koichestvo parametrov

package com.company;  
  
public class Cat {  
 public int height = 110;  
 public String name = "Default eto imia po ymolchaniu"; dlia raznoobrazia mozem dobavit6 polia s imenami   
  
 public void say (String name) {   
 System.*out*.print("Hello, " + name );  
  
 }  
 public Cat () {} eto konstryktor po ymolchaniu  
 public dlia sozdania konstryctora propisivaem modifikator dostypa Cat potom prorisivaem tochnoe nazvanie vashego klasa (int h v konstryktore mi mozem peredovat6 nekotorie parametric naprimer visota, String n mi propisiven dlia obwego znacheniia I dlia vseh parametrov ) nyzno obezatel6no dopisat6 skobki dlia konstryktora {  
 height = h; mi sozdaem znachenie dlia peremennoi vverhy public int height = 110;  
 name = n;  
 }  
 public Cat (String n) {  
 name = n;  
 }  
}

package com.company;  
  
public class Cat {  
 public int height = 110;  
 public String name = "Default";  
  
 public void say (String name) {  
 System.*out*.print("Hello, " + name );  
  
 }  
 public Cat () {}  
 public Cat (int h, String n) {  
 height = h;  
 name = n;  
 }  
 public Cat (String n) {  
 name = n;  
 }  
}

ob6ekti konstryktor pomogaet ob6ektam s parametrami

package com.company;  
import java.util.Scanner;  
  
  
public class Main {  
  
 public static void main(String[] args) {  
 Cat victor = new Cat(120, konstryktor kotorii prinimaet 1 parametr "Victor");  
 victor.say(" rubi ");  
 System.*out*.println(victor.name + "\ dabi skobki bil prostim simvolom nyzno postavit6 flew6 's height is " + victor.height);  
 Cat myrzik = new Cat (); eto parametric dlia konstryktora po ymolchaniu no on bydet brat6 parametry s obwego  
 myrzik.say ( " yrodi ");  
 System.*out*.println(myrzik.height);  
 Cat suka = new Cat ("suka");  
 System.*out*.println(suka.name);  
 }

package com.company;  
import java.util.Scanner;  
  
  
public class Main {  
  
 public static void main(String[] args) {  
 Cat victor = new Cat(120, "Victor");  
 victor.say(" rubi ");  
 System.*out*.println(victor.name + "\'s height is " + victor.height);  
 Cat myrzik = new Cat ();  
 myrzik.say ( " yrodi ");  
 System.*out*.println(myrzik.height);  
 Cat suka = new Cat ("suka");  
 System.*out*.println(suka.name);  
 }

NASLEDOVANIE KLASSOV

Kogda mi delaem naslednikom dochernii klass

package com.company;  
import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) {  
 Student hollyman = new Student(65, "XREN", 2);  
 hollyman.tell();  
 Student prinduk = new Student(63, "VASIA", 3);  
 prinduk.tell();  
 }

package com.company;  
import java.util.Scanner;  
  
  
public class Main {  
  
 public static void main(String[] args) {  
 Student hollyman = new Student(65, "XREN", 2);  
 hollyman.tell();  
 Student prinduk = new Student(63, "VASIA", 3);  
 prinduk.tell();  
 }

package com.company;  
public class Cat {  
 public int height = 110;  
 public String name = "Default";  
  
 public void say (String name) {  
 System.*out*.print("Hello, " + name );  
  
 }  
 public Cat () {}  
 public Cat (int h, String n) {  
 height = h;  
 name = n;  
 }  
 public Cat (String n) {  
 name = n;  
 }  
}

package com.company;  
  
public class Cat {  
 public int height = 110;  
 public String name = "Default";  
  
 public void say (String name) {  
 System.*out*.print("Hello, " + name );  
  
 }  
 public Cat () {}  
 public Cat (int h, String n) {  
 height = h;  
 name = n;  
 }  
 public Cat (String n) {  
 name = n;  
 }  
}

Dlia togo chtobi etot klass stal naslednikom nyzno dopisat6 EXTENDS inazvanie klassa

package com.company;  
public class Student extends Cat {  
 int course = 1;  
 public Student (int h, String n, int cuorse){  
 super (h, n);  
 this.course = cuorse;  
 }  
 void tell () {  
 System.*out*.println("your Height " + super.height);  
 System.*out*.println("Name " + super.name);  
 System.*out*.println("course " + course);  
 }  
}

package com.company;  
  
public class Student extends Cat {  
 int course = 1;  
  
 public Student (int h, String n, int cuorse){  
 super (h, n);  
 this.course = cuorse;  
 }  
  
 void tell () {  
 System.*out*.println("your Height " + super.height);  
 System.*out*.println("Name " + super.name);  
 System.*out*.println("course " + course);  
 }  
}