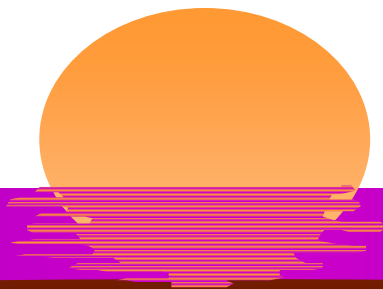


CS203 - Өгөгдлийн бүтэц 2020-2021

А. Хүдэр
КУ-ы салбарын
дэд профессор



Хичээлийн зорилго



- ♦ Өгөгдлийн бүтэц : дүрслэх, удирдах
- ♦ Бүх програм өгөгдөлтэй харьцдаг
- ♦ Тэгэхээр програм өгөгдлийг ямар нэг байдлаар дүрслэнэ.
- ♦ Өгөгдлийг удирдахад алгоритм хэрэгтэй

Хичээлийн зорилго




- ♦ Өгөгдлийг удирдах програмыг хөгжүүлэхэд алгоритмын зохиомжийн аргууд хэрэгтэй
- ♦ Өгөгдлийн бүтэц, алгоритм бол компьютерийн ухааны суурь юм

Сурах бичиг



- ♦ Sahni Sartaj. Data Structures, Algorithms and Applications in Java. 2000. ISBN 0-07-109217-X
- ♦ Сургуулийн номын санд 1 хувьтай.
- ♦ Хичээлийн Веб хуудсанд шаардлагатай бүлгүүдийн хувилсан хуулбарыг pdf форматаар тавьна.
- ♦ Сурах бичигт орсон жишээний эх код мөн zip форматаар тавигдана.

Тавигдах шаардлага

- ♦ Java 
- ♦ Алгоритмын үнэлгээ
 - Том O, тета, омега тэмдэглэгээ



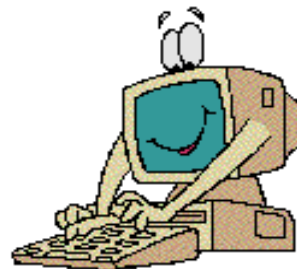
Веб хуудас



- ♦ elearn.sict.edu.mn
- ♦ Хичээлийн хөтөлбөр, лекцийн үзүүлэн, бие даалт, сурах бичиг, эх код гээд хэрэгтэй бүхнээ олж болно

Бие даалт

- ♦ Удирдамжийг удирдлага болгох
- ♦ Хугацаандаа өгөх
- ♦ Хамгаалалт дээр програмаа тайлбарлана
- ♦ Нийт 4 бие даалт





Лаборатори



- ♦ Лабораторийн хичээл дээр багш жишээ програм бичиж үзүүлнэ
- ♦ Багшаас ойлгоогүй зүйлээ асууна
- ♦ Лабораторын ажиллах орчин - Eclipse (www.eclipse.org) эсвэл BlueJ

Үнэлгээ

- ♦ 30 - улирлын шалгалт
- ♦ 20 - завсрын шалгалт 2 удаа
- ♦ 20 - бие даалт (5 оноо х 4 удаа)
- ♦ 14 – лабораторийн ажил (2 оноо х 7 удаа)
- ♦ 16 – лаб, лекцийн идэвх, ирц (1 х 16 долоо хоног)

Java



Програмын шинжүүд:

- ♦ Зөв үү?
- ♦ Програмыг унших, кодыг ойлгох амархан уу?
- ♦ Програм хир баримтжуулагдсан?
- ♦ Өөрчлөлт оруулах амархан уу?
- ♦ Хир ой шаардагдах вэ?
- ♦ Хир удаан ажилах вэ?
- ♦ Програм хир нийтлэг вэ?
- ♦ Өөрчлөлтгүй өөр машин дээр ажиллах уу?



Програмын бүтэц

Программ класс -> гишүүд (өгөгдөл, арга)

Бие даасан программ -> main()

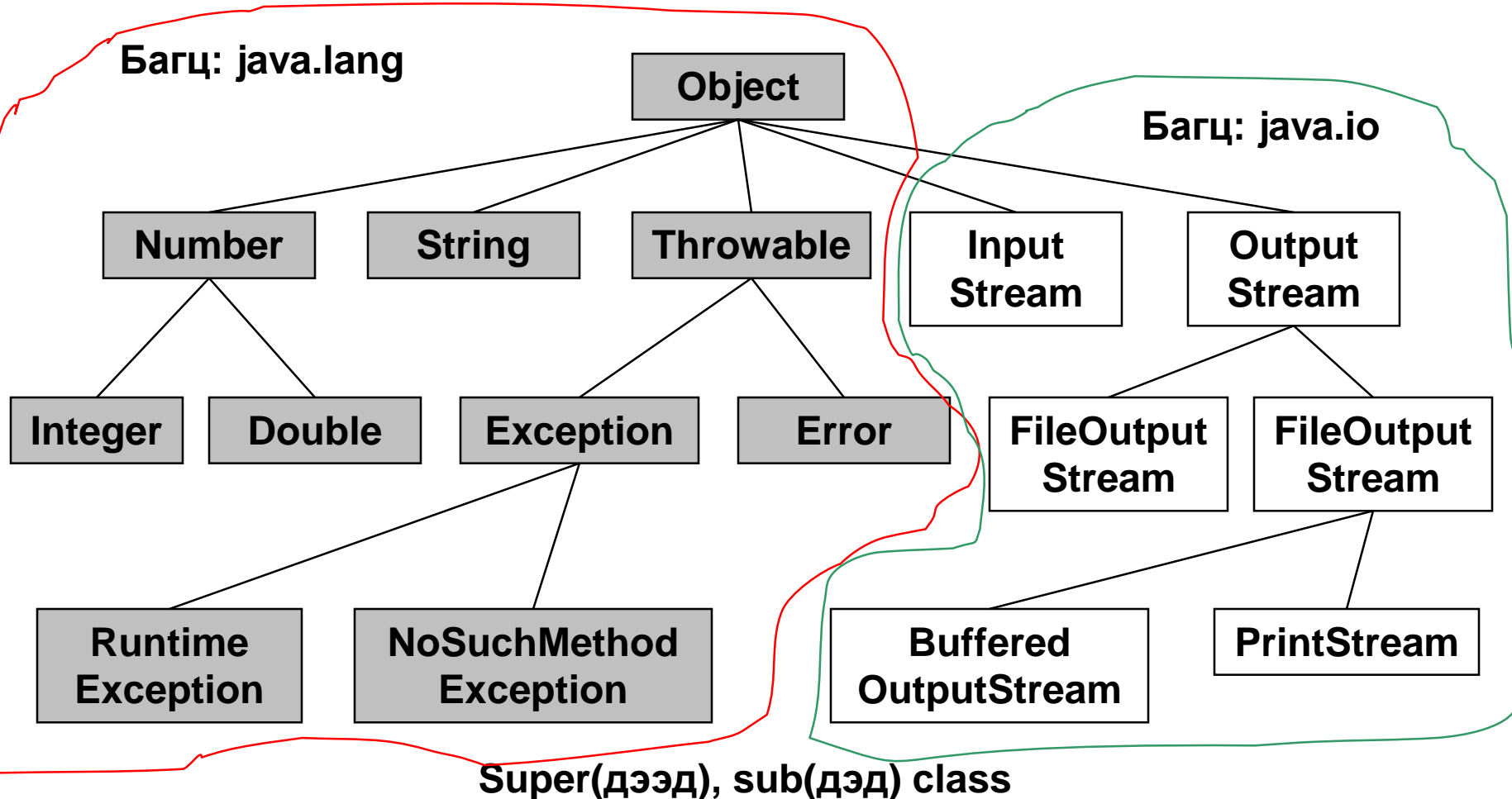
Багц: стандарт багц: java.awt, java.io, java.lang(Integer, String, Boolean ...), java.util

Сурах бичиг: applications, dataStructures, wrappers, exceptions, utilities, misc.

System.out.println("Welcome"); -> System: java.lang багцын класс, out: гишүүн (төрөл нь java.io багцын PrintStream)

Классыг импортлох: import java.io.*

Программын бүтэц



`public class Welcome extends NemaOfSuperClass`

`public final class Welcame`



Java Виртуаль машин, Javadoc

Java хөрвүүлэгч: .java (эх код) -> .class
(байткод)

Java виртуаль машин:

- + машинаас хамаарахгүй, цомхон, найдвартай
- удаан

Javadoc: /** тайлбар */, @param,
@throws, @return ... -> HTML page

Өгөгдлийн төрөл



Төрөл	Утга	Бит	Хүрээ
boolean	False	1	[false, true]
byte	0	8	[-128, 127]
char	\u0000	16	[\u0000, \uFFFF]
double	0.0	64	[+/-4.9e-324, +/-1.8e308]
float	0.0	32	[+/-1.4e-45, +/-3.4e38]
int	0	32	[-2147483648, 2147483647]
long	0	64	+/-9.2e17
short	0	16	[-32768, 32767]

Byte, Integer, Boolean, String -> Объект



Гишүүн функц

```
public static int abc(int a, int b, int c) {  
    return a+b*c+b/c}
```

- ♦ Параметр: зарласан, жинхэнэ
- ♦ Утга дамжина -> дотоод хувьсагч шиг
- ♦ Давхар ачаалах: `public static float abc(...) {...}`

Онцгой тохиолдол



Дандаа алдаа байдаггүй!

- ◆ Шидэх:

- ◆ `if(a<0 || b<0 || c<0)`

- `throw new IllegalArgumentException`
`("All parameters must be >=0");`

- ◆ Барьж авч, боловсруулах:

- ◆ `try {System.out.Println(abc(2,-3,4));`

- `} catch (IllefalArgumentException e) {`
`System.out.Println("Positive paraneters are`
`needed!");}` `finaly { ... }`

Шинэ төрөл



```
public class Currency {  
    public static final boolean PLUS=true, MINUS=false;  
    private boolean sign;  
    private long dollars;  
    private byte cents;  
    public Currency(boolean sign, long dollars, byte cents) { ... }  
    public Currency() { this(PLUS,0L,(byte)0); }  
    public boolean getSign() { ... }  
    public long getDollars() { ... }  
    public void setSign(boolean sign) { ... }  
    public void setDollars(long dollars) { ... }  
    ...  
}
```



Шинэ төрөл

- ♦ Өгөгдөл гишүүн (тогтмол): `public static final boolean PLUS=true, MINUS=false;`
- ♦ Өгөгдөл гишүүн (тохиолдол): `private boolean sign; private long dollars; private byte cents;`
- ♦ Арга гишүүн (байгуулагч): `public Currency(...) { ... } public Currency() { ... }`
- ♦ Арга гишүүн: `public boolean getSign() { ... } public long getDollars() { ... } public void setSign(boolean sign) { ... } public void setDollars(long dollars) { ... }`

Хандалтын тодорхойлогч



Тодорхойлогч	Гишүүний харагдах хүрээ
-	Тухайн багцын классууд
private	Тодорхой C класст
protected	Тухайн багцын классууд, өөр багцын C дэд класст
public	Бүх багцын бүх класст

Удамшил, гишүүн функцийг орлуулах(солих)



Шинэ класс дээд классаасаа удамшдаг:
тохиолдолд гишүүн өгөгдлийн хуулбар,
гишүүн функцууд удамшдаг.

Currency класс Object классаас удамшсан:

```
public boolean equals(Object obj); public  
String toString(); -> Object класс ->  
if(currA.equals(currB) { ... }
```

Currency.equals() -> Object.equals() –г орлоно



Интерфейс

```
public interface LinkedList {  
    public boolean isEmpty();  
    public int size();  
    public Object get(int index);  
    public int indexOf(Object elem);  
    public Object remove(int index);  
    public void add(int index, Object obj);  
    public String toString();  
}
```

Хаягдал цуглуулах - Garbage Collection



```
int [] a = new int[50000];
```

```
Currency c = new Currency();
```

```
a = null;
```

```
c = null;
```

Хэрвээ a[] int биш Currency байсан бол 7Mb ой хэрэгтэй болно.

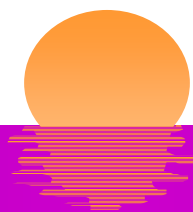
Рекурс



```
public static int factorial(int n) {  
    if (n <= 1)  
        return 1;  
    else  
        return n * factorial(n-1);  
}
```

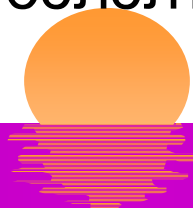
Эрэмбэлэлт

- ♦ $a[0], a[1], \dots, a[n-1]$ массивын элементүүдийг өсөхөөр эрэмбэлэх.
 $a[0] \leq a[1] \leq \dots \leq a[n-1]$ болгох.
- ♦ $8, 6, 9, 4, 3 \Rightarrow 3, 4, 6, 8, 9$



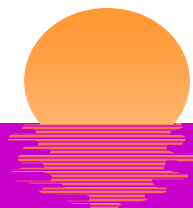
Эрэмбэлэлтийн аргууд

- ♦ Оруулж эрэмбэлэх
- ♦ Бөмбөлгөн эрэмбэлэлт
- ♦ Сонгон эрэмбэлэх
- ♦ Тоолж эрэмбэлэх
- ♦ Сэгсэрч эрэмбэлэх
- ♦ Шеллийн эрэмбэлэлт
- ♦ Пирамидан эрэмбэлэлт
- ♦ Нэгтгэн эрэмбэлэх
- ♦ Хурдан эрэмбэлэлт



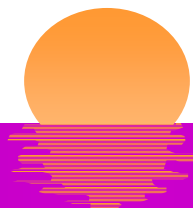
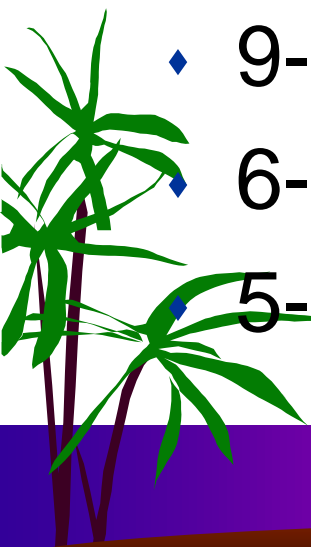
Элемент оруулах

- ♦ Эрэмбэлэгдсэн дараалал өгөгдөхөд шинэ элемент оруулах
- ♦ Өгсөн: 3, 6, 9, 14
- ♦ Оруулах: 5
- ♦ Үр дүн: 3, 5, 6, 9, 14



Элемент оруулах

- ♦ 3, 6, 9, 14 5-г оруулах
- ♦ Шинэ элементийг (5) сүүлийнхтэй (14) харьцуулах
- ♦ 14-г баруун тийш болгох 3, 6, 9, , 14
- ♦ 9-г баруун тийш болгох 3, 6, , 9, 14
- ♦ 6-г баруун тийш болгох 3, , 6, 9, 14
- ♦ 5-г оруулах 3, 5, 6, 9, 14



Элемент оруулах

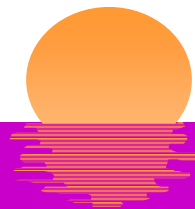
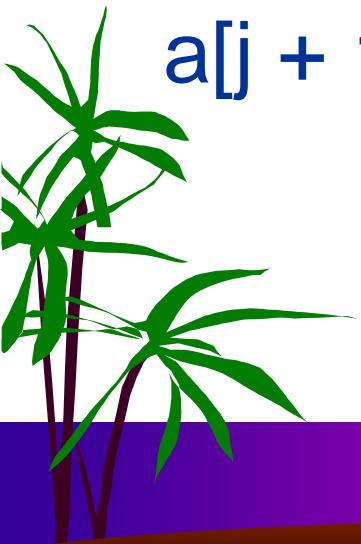
// t-г $a[0:i-1]$ руу оруулах

int j;

for (j = i - 1; j >= 0 && t < a[j]; j--)

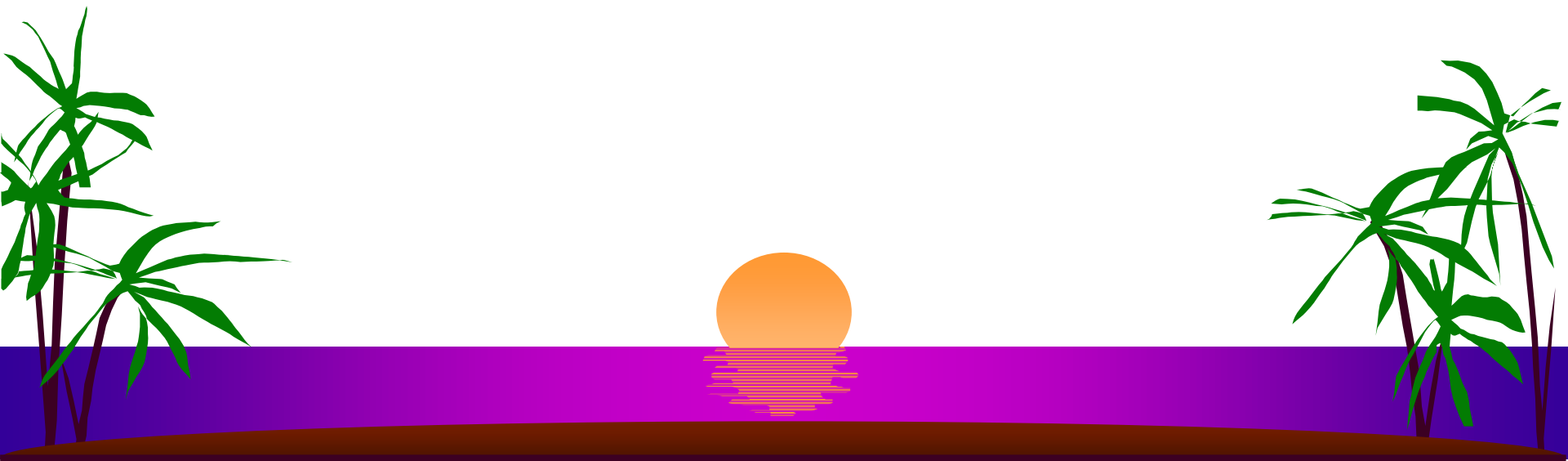
$a[j + 1] = a[j];$

$a[j + 1] = t;$



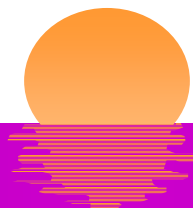
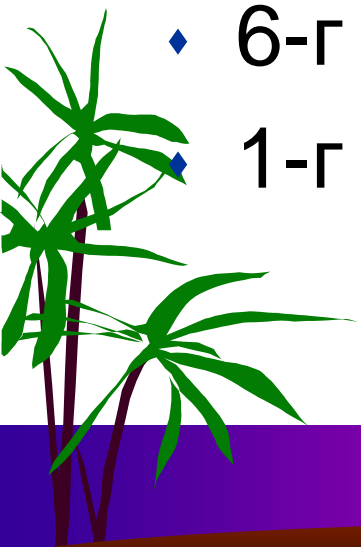
Элемент оруулах

- ◆ Дараалал нь эхлээд 1 гэсэн урттай байна
- ◆ Үлдсэн элементүүдийг нэг нэгээр нь оруулна



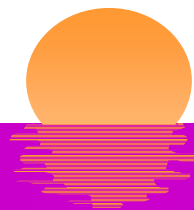
Оруулж эрэмбэлэх

- ♦ 7, 3, 5, 6, 1 – г эрэмбэлэх
- ♦ 7 гэсэн дараалал руу 3-г оруулах \Rightarrow 3, 7
- ♦ 5-г оруулах \Rightarrow 3, 5, 7
- ♦ 6-г оруулах \Rightarrow 3, 5, 6, 7
- ♦ 1-г оруулах \Rightarrow 1, 3, 5, 6, 7



Оруулж эрэмбэлэх

```
for (int i = 1; i < a.length; i++)  
{// a[i]-г a[0:i-1] руу оруулах  
  // оруулах кодыг энд бичнэ  
}
```



Оруулж эрэмбэлэх

```
for (int i = 1; i < a.length; i++)  
{// a[i]-г a[0:i-1] руу оруулах  
    int t = a[i];  
    int j;  
    for (j = i - 1; j >= 0 && t < a[j]; j--)  
        a[j + 1] = a[j];  
    a[j + 1] = t;  
}
```

