

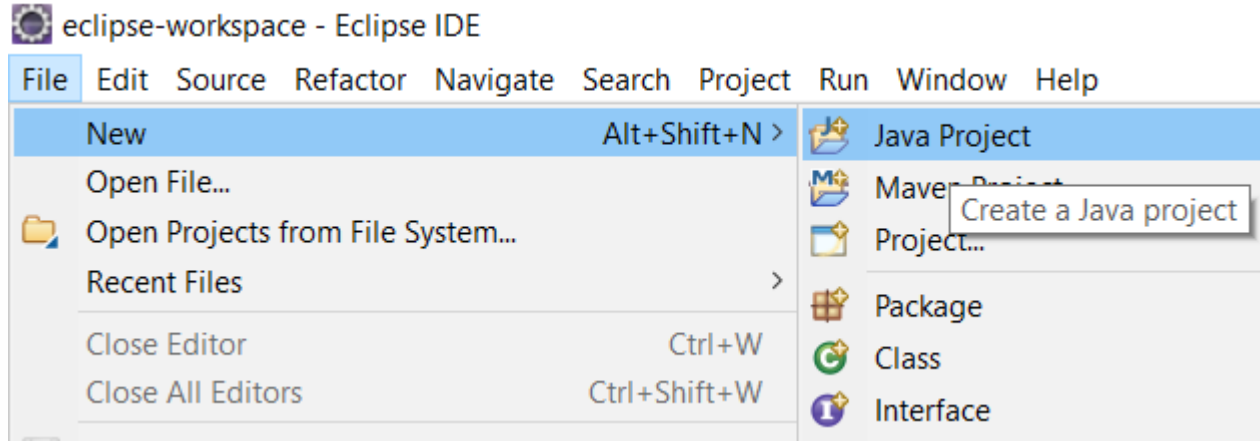
NESNE TABANLI PROGRAMLAMA

JAVA

İÇİNDEKİLER

- ▶ Yeni Proje Açma
- ▶ Veriler
- ▶ Veri Tipleri
- ▶ Değişkenler
- ▶ Operatörler

Yeni Proje Açma



Yeni Proje Açma

- Proje İsmi Verilir
- Finish

New Java Project

Create a Java Project

Discouraged module name. By convention, module names usually start with a lowercase letter

Project name: NesneYonelimliYazilim

☒ Use default location

Location: C:\Users\nmacit\eclipse-workspace\NesneYonelimliYazilim [Browse...](#)

JRE

☒ Use an execution environment JRE: JavaSE-21 [Configure JREs...](#)

☐ Use a project specific JRE: jre

☐ Use default JRE 'jre' and workspace compiler preferences

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets [New...](#)

Working sets: [Select...](#)

Module

☒ Create module-info.java file

Module name:

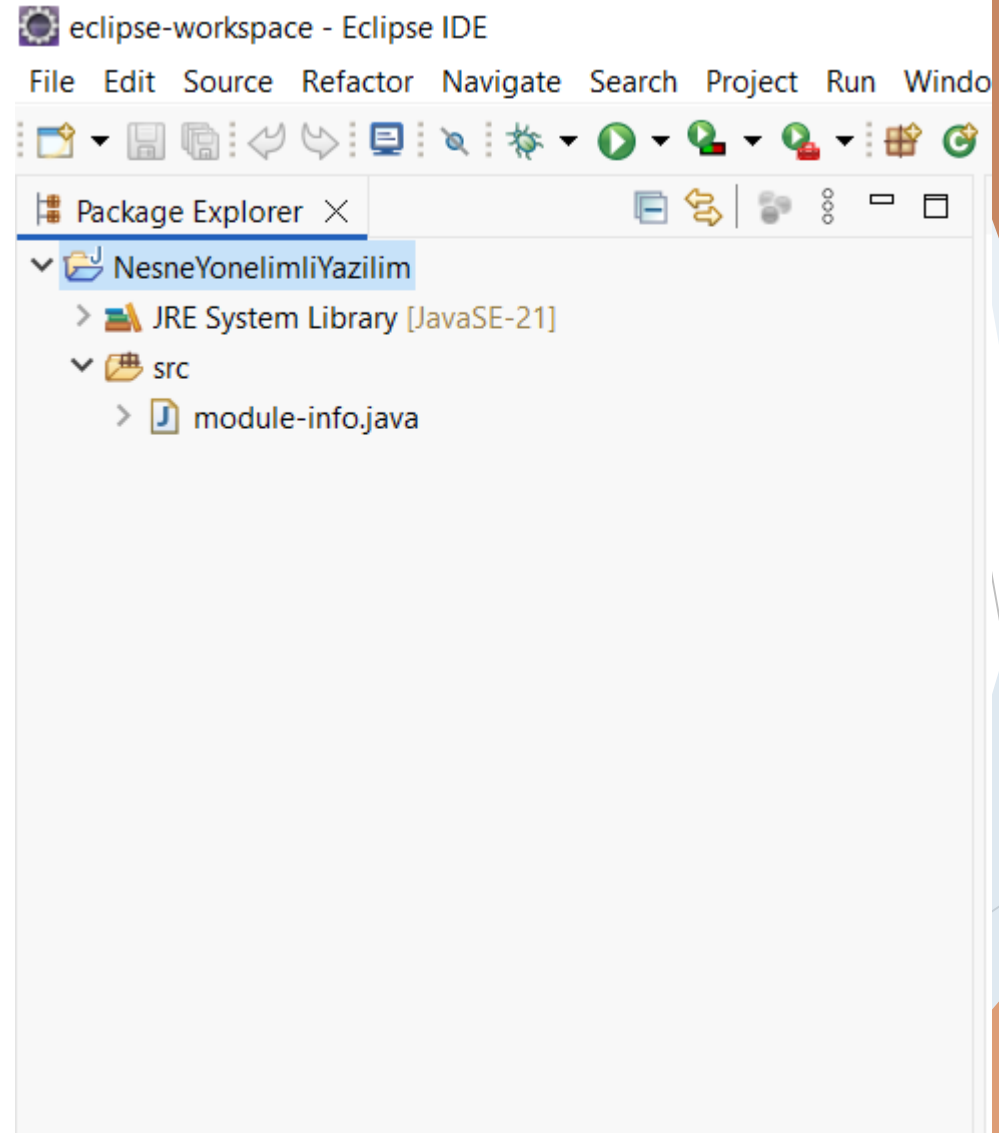
☒ Generate comments

module name will be "NesneYonelimliYazilim" (if no module is specified, then project name will be used as module name)

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

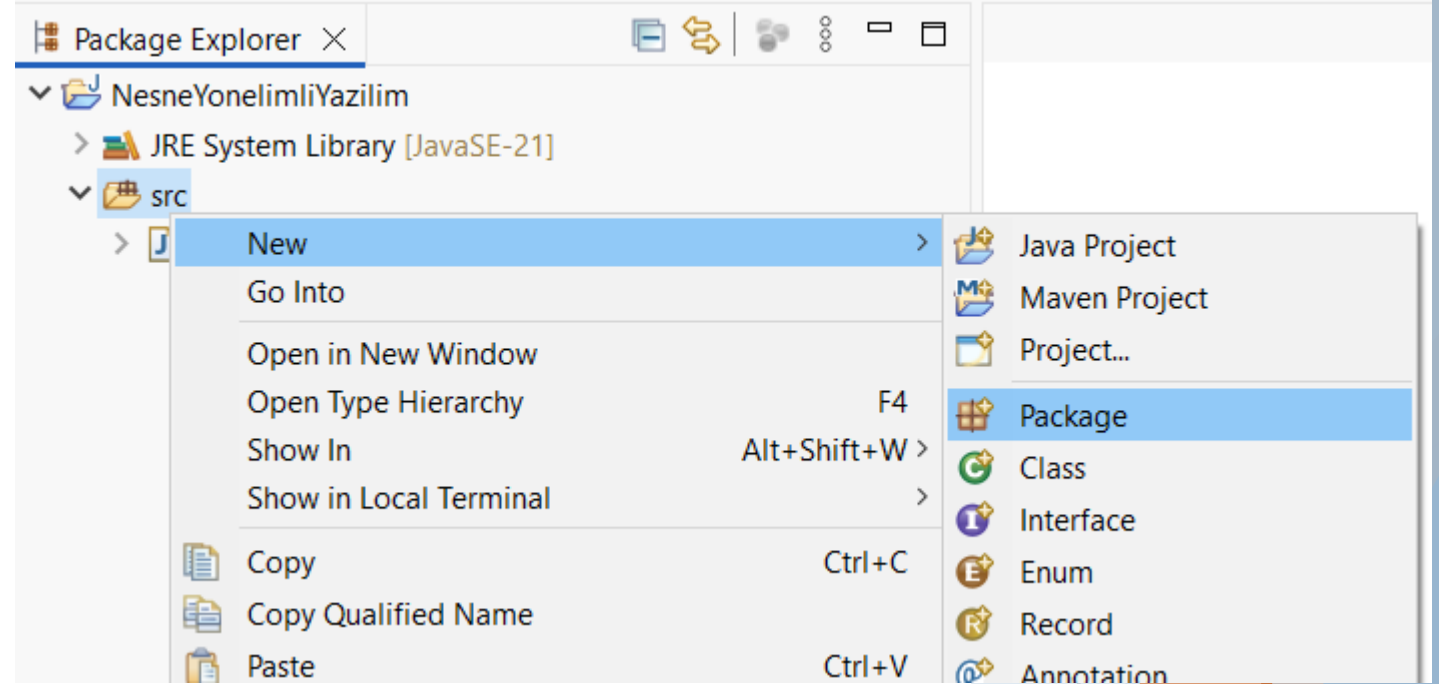
Yeni Proje Açma

- ▶ Proje altında default kütüphanelerinizi
- ▶ Src adında bir klasörünüz bulunur.



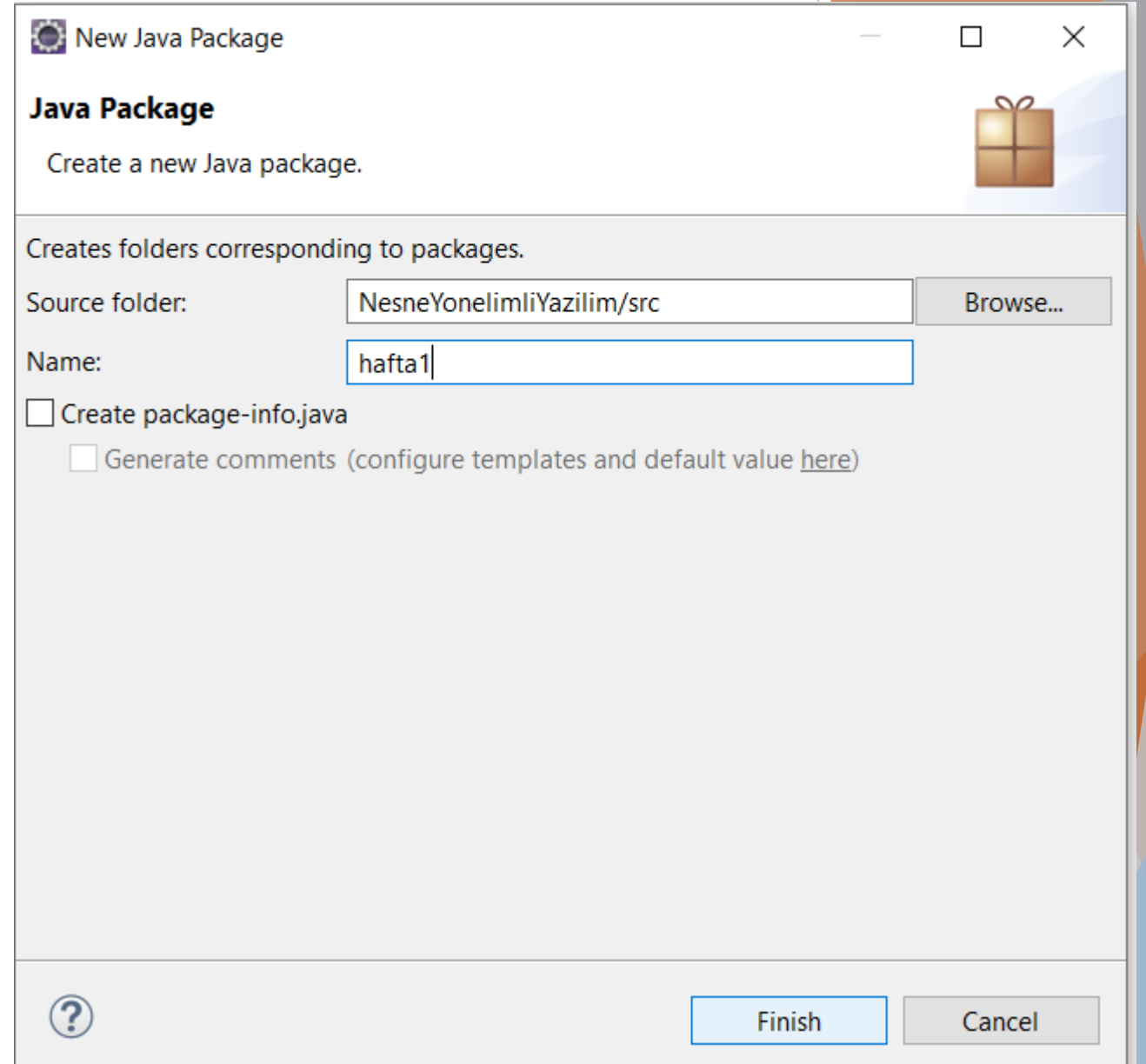
Yeni Proje Açma

- Src üzerinde sağ tık ile yeni bir package (klasör) oluşturulur.



Yeni Proje Açma

- Src üzerinde sağ tık ile yeni bir package (klasör) oluşturulur.
Hafta1 adını verdik
Finish



New Java Package

Java Package
Create a new Java package.

Creates folders corresponding to packages.

Source folder:

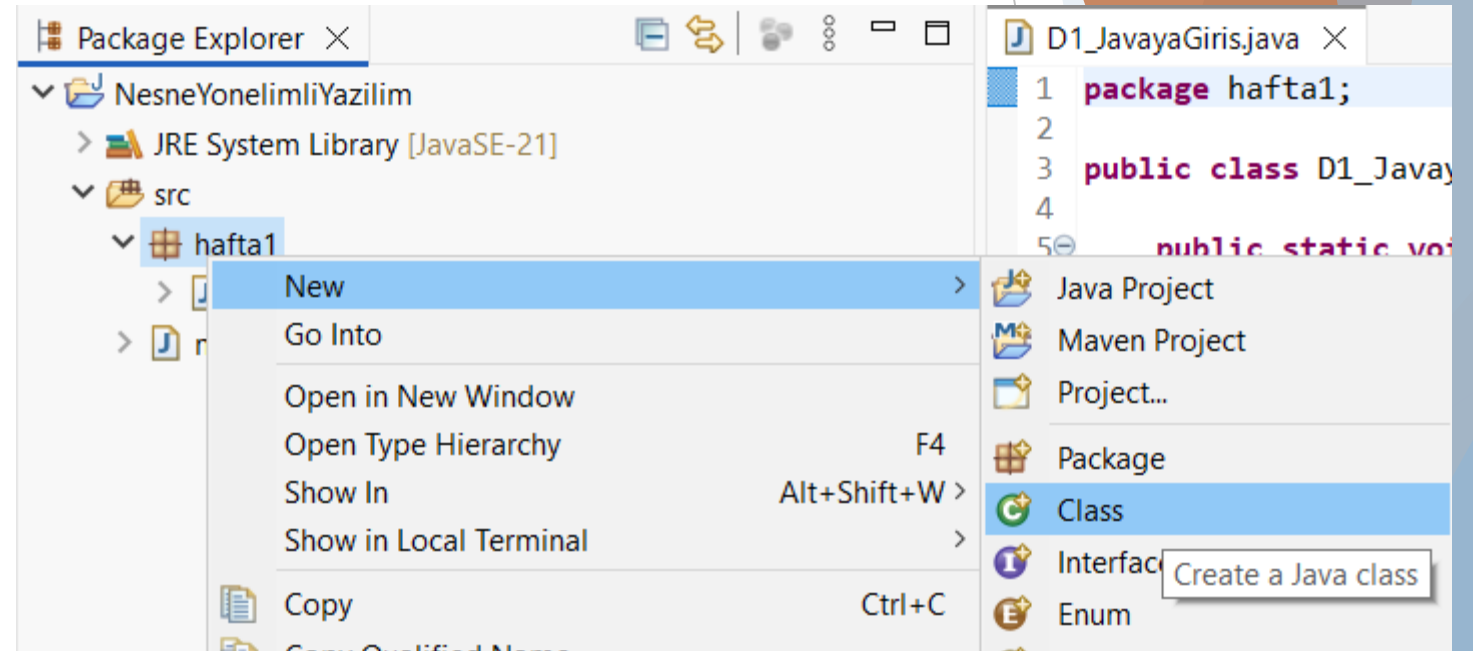
Name:

☐ Create package-info.java

☐ Generate comments (configure templates and default value [here](#))

Yeni Proje Açma

- hafta1 üzerinde sağ tık ile yeni bir class (sınıf) oluşturulur.



Yeni Proje Açma

- ▶ Name kısmına adını yazdık
- ▶ DİKKAT sınıf isimleri büyük harfle başlar
- ▶ Sınıf ismini belirledik
- ▶ main metodu eklemek için Aşağıda ilgili kutucuğa tik koyduk.
- ▶ Finish

New Java Class

Java Class

Create a new Java class.

Source folder: NesneYonelimliYazilim/src Browse...

Package: hafta1 Browse...

☐ Enclosing type: Browse...

Name: D1_JavayaGiris

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

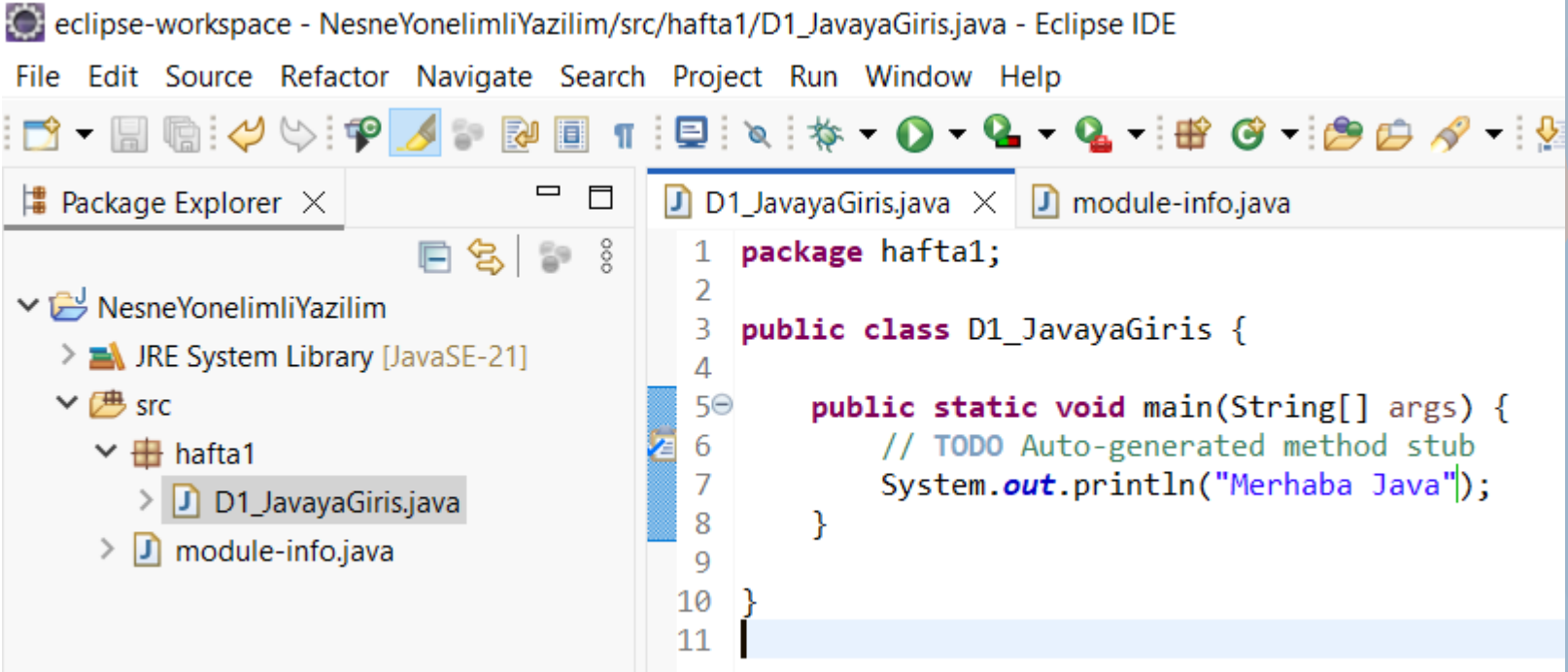
☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Finish Cancel

Yeni Proje Açma

- ▶ Package Explorer penceresi projenize ait klasörleri görebileceğiniz alandır.
- ▶ Proje Adı NesneYonelimliYazilim
- ▶ JRE System Library Proje içine dahil edilen kütüphaneler
- ▶ Src klasörü projenizin kaynak kodlarının bulunduğu kısımdır.
- ▶ Src klasörü altında hafta1 adına bir klasör daha oluşturup
- ▶ İçerisine yeni bir class tanımladık.

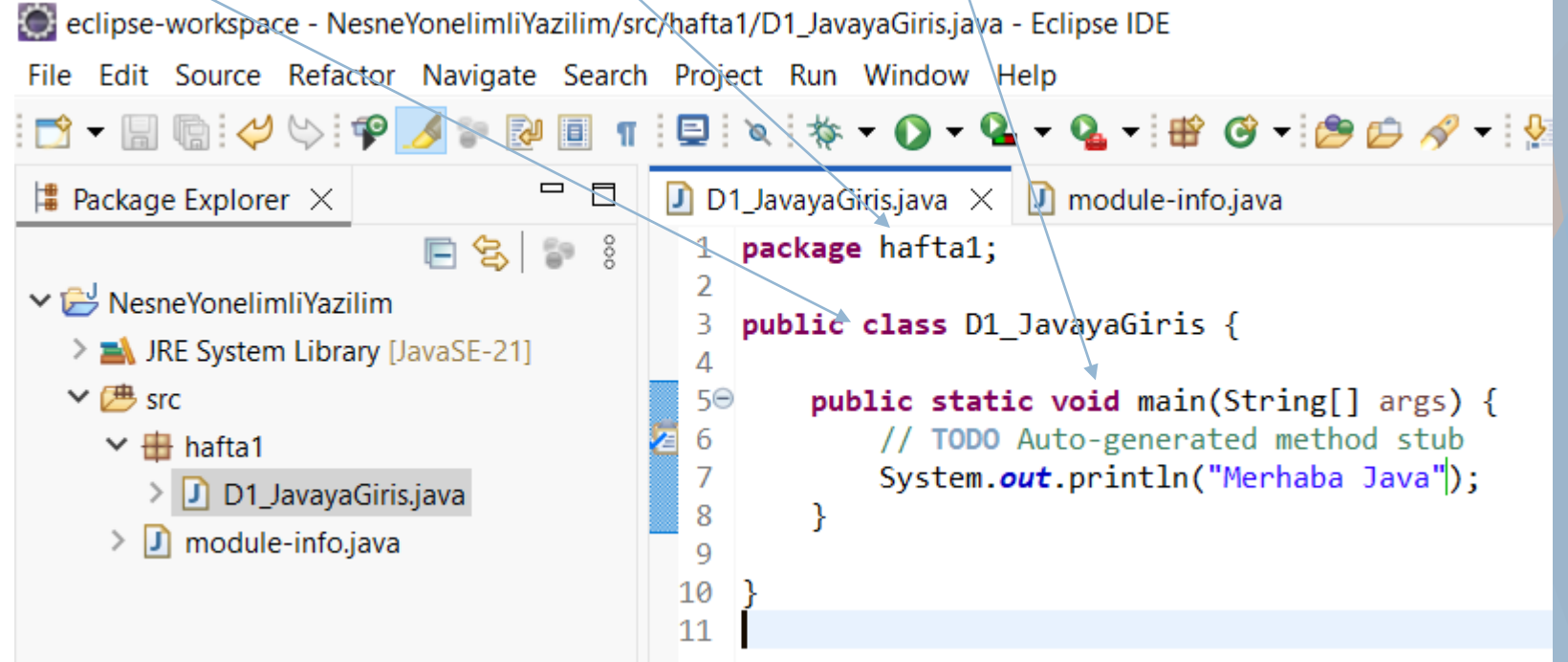


The screenshot shows the Eclipse IDE interface. The title bar reads "eclipse-workspace - NesneYonelimliYazilim/src/hafta1/D1_JavayaGiris.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations and development. The Package Explorer on the left shows the project structure: NesneYonelimliYazilim (expanded) contains JRE System Library [JavaSE-21] and src (expanded). Under src, there is a folder named hafta1 (expanded), which contains two files: D1_JavayaGiris.java (selected) and module-info.java. The main editor area shows the code for D1_JavayaGiris.java, which is a Java class with a package declaration and a main method.

```
1 package hafta1;
2
3 public class D1_JavayaGiris {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("Merhaba Java");
8     }
9
10 }
11
```

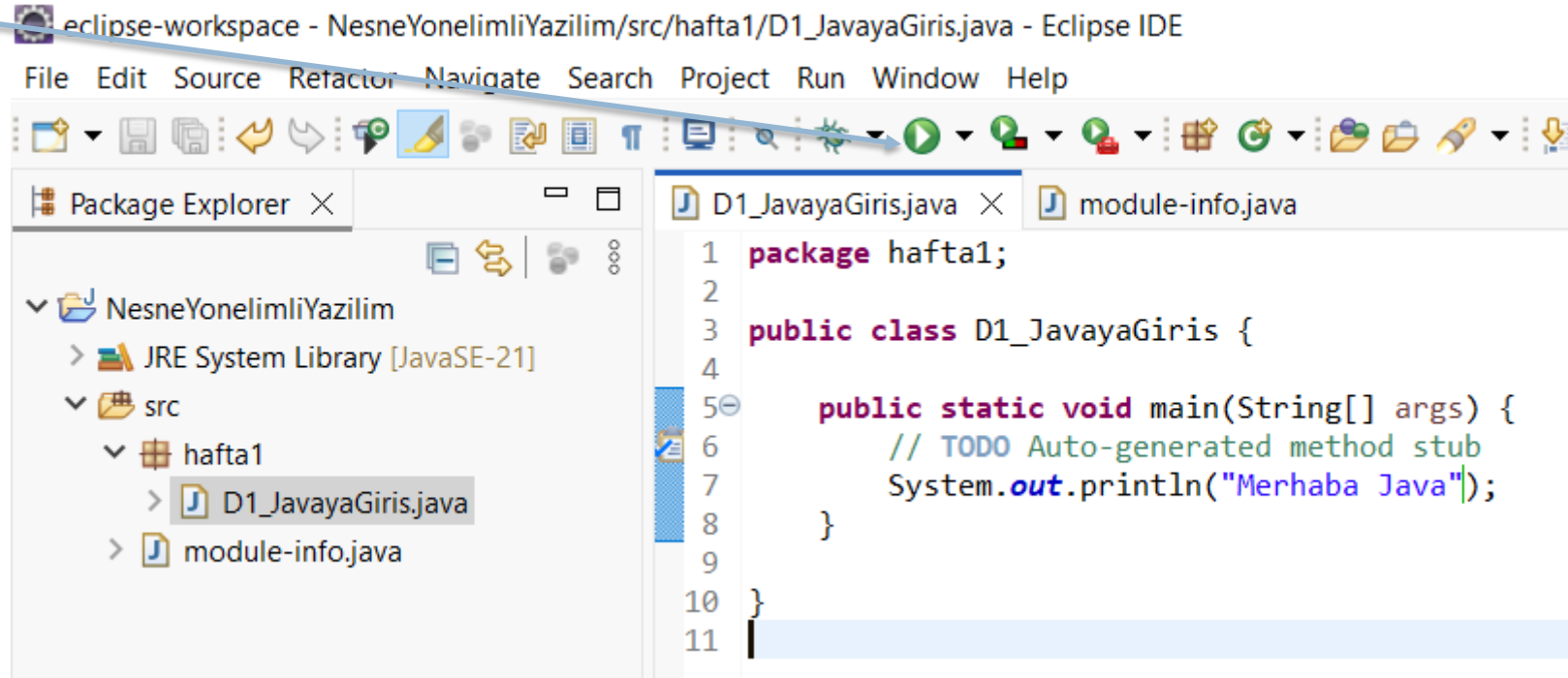
Yeni Proje Açma

- ▶ Bu main class ı açtığımızda sağ tarafta pencere açıldı.
- ▶ Bu pencerede deki ilk kodlar
- ▶ Bu sınıfın hangi klasör altında olduğunu gösterir.
- ▶ Bu sınıfın adını gösterir.
- ▶ Program çalışır çalışmaz ilk derlenecek kodlar da main fonksiyonuna yazılır.



Ekrana Yazı Yazdırmak

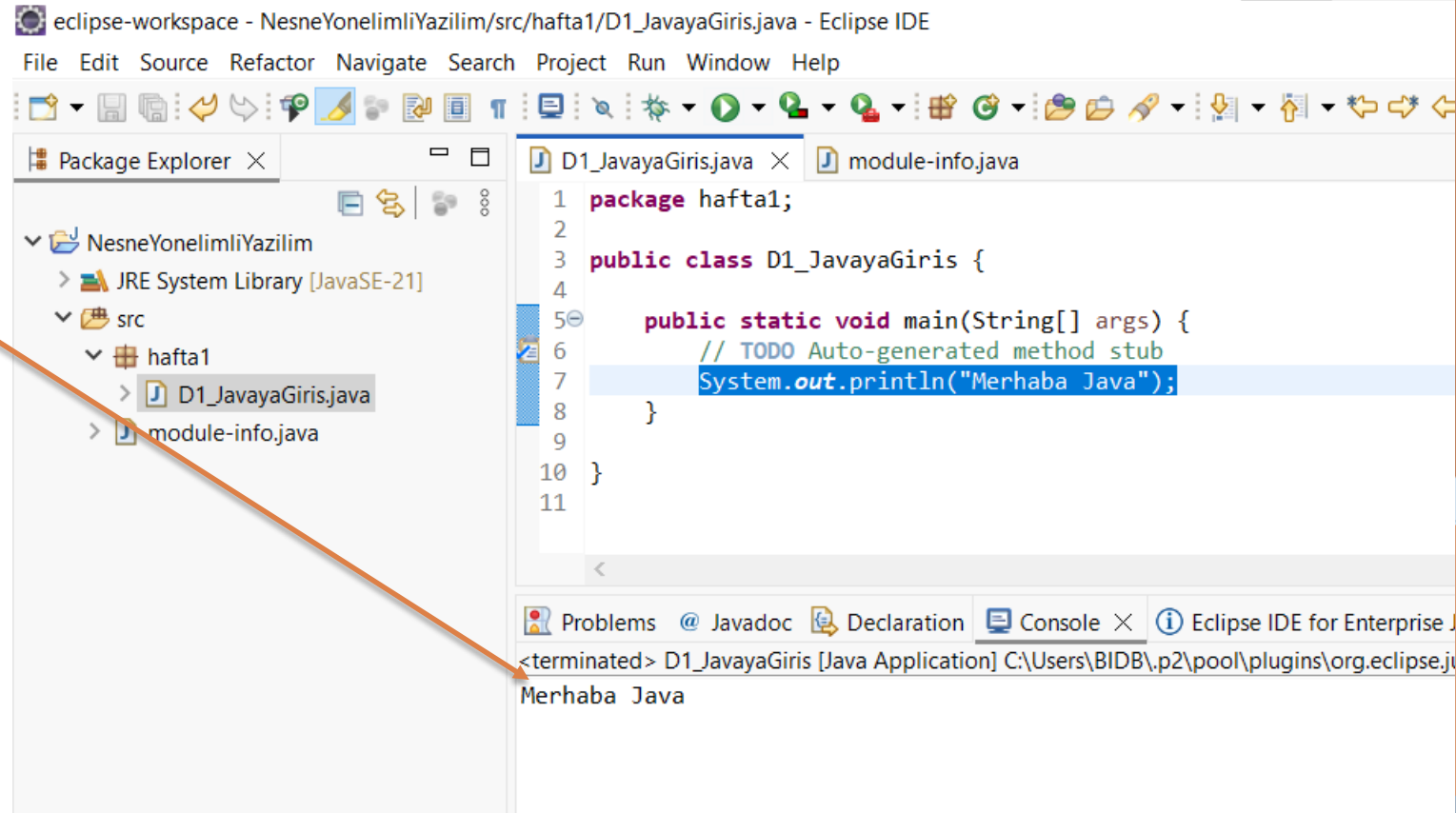
- ▶ `System.out.println("Merhaba Java");`
- ▶ Run butonuna tıklayın



Ekrana Yazı Yazdırmak

- ▶ `System.out.println("Merhaba Java");`
- ▶ Run butonuna tıklayın
- ▶ Alt panelde console

Ekranında ekran çıktısını
Görüyor olacaksınız.



VERİLER

İlkel Veri Tipleri Primitive Data Types

İlkel Olmayan Veri Tipleri Non Primitive Data Types

Veriler



Giriş Birimlerinden Girilebilecek Veriler

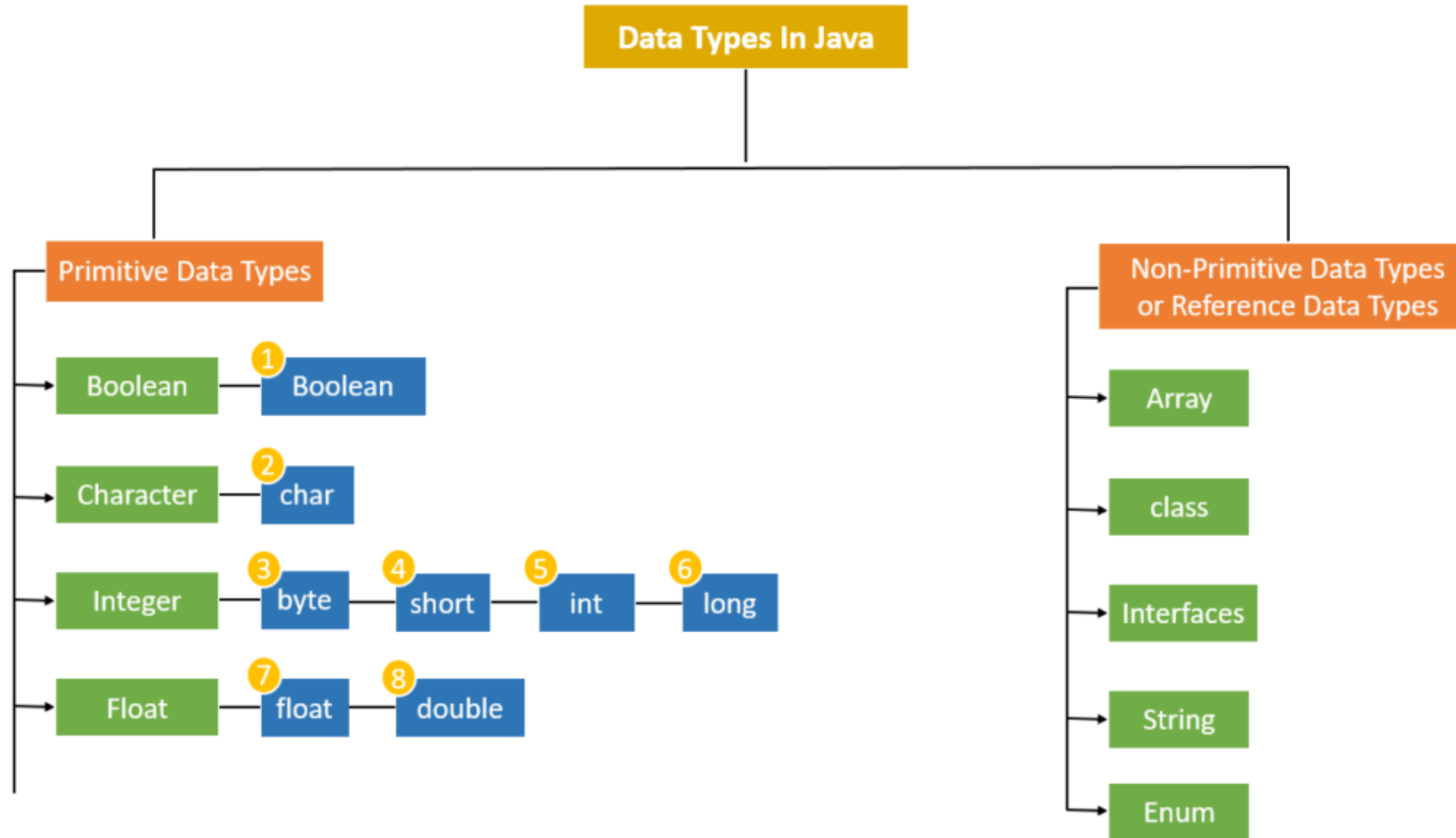
➤ SAYILAR

- Tam sayılar: $-\infty \dots -100 \dots -2, -1, 0, 1, 2, 3 \dots +\infty$
- Ondalıklı sayılar :
 $-\infty \dots -2, -1, 0, 0.2, 0.7, 0.79, 0.8, 1, 1.7, 2, \dots +\infty$
- Boolean Sayılar: true(1), false(0)

➤ SÖZEL İFADELER

- Tek Karakterlik : 'N', 'a', '!', '*', ...
- Birden fazla karakterlik: "Naciye", "nama22", "123", "sezikli*-"

JAVA DA VERİ TİPLERİ

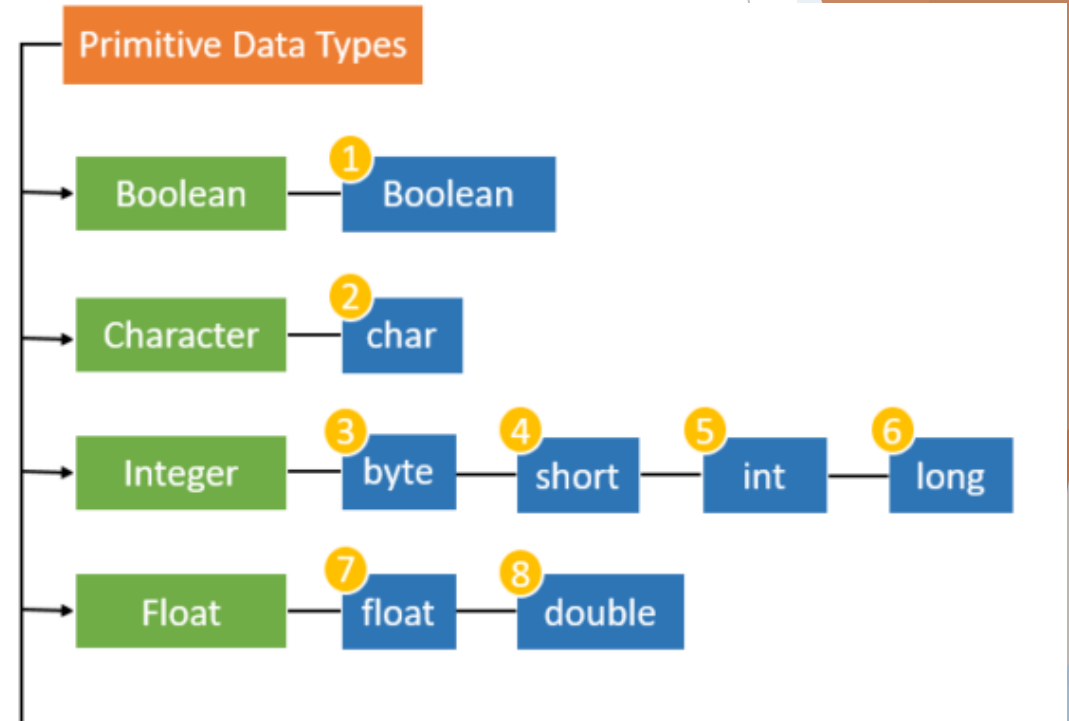


JAVA DA VERİ TİPLERİ

- Java, **tip güvenliği yüksek (strongly typed)** bir programlama dilidir. Bu yüzden tanımlanan her değişken mutlaka belirli bir veri tipiyle ilişkilendirilir.
- Programcı, değişkenlere uygun türde değerler atadığından emin olmalıdır; aksi durumda program hata verebilir.

Primitive Data Types(İlkel Veri Tipleri)

- ▶ İlkel veri tipleri bellekte doğrudan değerleri tutar
- ▶ Java'daki primitive veri tipleri java dilinin kendisi tarafından tanımlanmıştır, bu nedenle yeni bir primitive tip oluşturulamaz.
- ▶ Primitive veri tiplerinin isimleri daima **küçük harflerle** yazılır.
- ▶ Her primitive tip, türüne bağlı olarak bellekte farklı miktarda yer kaplar. Kapladığı alan değişmez.
- ▶ Primitive tipler yalnızca atanan değeri saklar, başka bir bilgi içermez.



Veri Tipleri

➡ SAYILAR

➡ Tam sayılar: $-\infty \dots -100 \dots -2, -1, 0, 1, 2, 3 \dots +\infty$

Byte, Short, Integer, Long

Ondalıklı sayılar : $-\infty \dots -2, -1, 0, 0.2, 0.7, 0.79, 0.8, 1, 1.7, 2, \dots +\infty$ **Float, Double**

Boolean Sayılar: **true(1), false(0), Boolean**

➡ SÖZEL İFADELER

➡ Tek Karakterlik : 'N', 'a', '!', '*', ... **Char**

➡ Birden fazla karakterlik: "Naciye", "nama22", "123", "sezikli*-" **String**

Veri Tipleri->TAM SAYILAR

<u>Java Tipi</u>	<u>class</u>	<u>Uzunluk</u>	<u>Değer Aralığı</u>
byte	Byte	8 BİT	0 ... 255
short	Short	16 BİT	-32 768 ... 32 767
int	Integer	32 BİT	-2 147 483 648 ... +2 147 483 648
long	Long	64 BİT	-2^{63} ... $2^{63} - 1$

Veri Tipleri->ONDALIKLI SAYILAR

<u>İlkel tip</u>	<u>class</u>	<u>Uzunluk</u>	<u>Değer Aralığı</u>
float	Single	32 BİT	$1.5 * 10^{-45}$ - $3.4 * 10^{38}$
double	Double	64 BİT	$5.0 * 10^{-324}$ - $1.7 * 10^{308}$

- Aksi istenmediğinde, Java kesirli sayıların double tipinden olduğunu varsayar (ön tanımlı, default tip).
- float a = 45.93F;
- **NOT :** Yukarıda gördüğünüz gibi sayıdan sonra “F” yazdık. Bu ne demek oluyor ? Java’da noktalı tiplerde varsayılan değer double’dır. Yani siz bir noktalı sayı yazdığınızda Java bunu double tipi olarak okur. Biz bunun float olduğunu anlatmak için sayının sonuna “f” veya “F” koymalıyız. Aksi halde hata alırız.
- Çift duyarlıdırlar.

Veri Tipleri->BOOLEAN SAYILAR

<u>İlkel tip</u>	<u>class</u>	<u>Uzunluk</u>	<u>Değer Aralığı</u>
boolean	Bool	1 BİT	True-false

Veri Tipleri->KARAKTER

<u>İlkel tip</u>	<u>class</u>	<u>Uzunluk</u>	<u>Değer Aralığı</u>
char	char	16 BİT	0-(2 ¹⁶ -1)

- Her tam sayı Unicode karakter setinde özel bir karaktere karşılık gelir

İkilik sistem	Sekizlik sistem	Onluk sistem	On altılık sistem	Karakter	İkilik sistem	Sekizlik sistem	Onluk sistem	On altılık sistem	Karakter	İkilik sistem	Sekizlik sistem	Onluk sistem	On altılık sistem	Karakter
010 0000	040	32	20	SP	100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	\$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n
010 1111	057	47	2F	/	100 1111	117	79	4F	O	110 1111	157	111	6F	o
011 0000	060	48	30	0	101 0000	120	80	50	P	111 0000	160	112	70	p
011 0001	061	49	31	1	101 0001	121	81	51	Q	111 0001	161	113	71	q
011 0010	062	50	32	2	101 0010	122	82	52	R	111 0010	162	114	72	r
011 0011	063	51	33	3	101 0011	123	83	53	S	111 0011	163	115	73	s
011 0100	064	52	34	4	101 0100	124	84	54	T	111 0100	164	116	74	t
011 0101	065	53	35	5	101 0101	125	85	55	U	111 0101	165	117	75	u
011 0110	066	54	36	6	101 0110	126	86	56	V	111 0110	166	118	76	v
011 0111	067	55	37	7	101 0111	127	87	57	W	111 0111	167	119	77	w
011 1000	070	56	38	8	101 1000	130	88	58	X	111 1000	170	120	78	x
011 1001	071	57	39	9	101 1001	131	89	59	Y	111 1001	171	121	79	y
011 1010	072	58	3A	[[:]]	101 1010	132	90	5A	Z	111 1010	172	122	7A	z
011 1011	073	59	3B	;	101 1011	133	91	5B	[111 1011	173	123	7B	{
011 1100	074	60	3C	<	101 1100	134	92	5C	\	111 1100	174	124	7C	
011 1101	075	61	3D	=	101 1101	135	93	5D]	111 1101	175	125	7D	}
011 1110	076	62	3E	>	101 1110	136	94	5E	^	111 1110	176	126	7E	~
011 1111	077	63	3F	?	101 1111	137	95	5F	_					

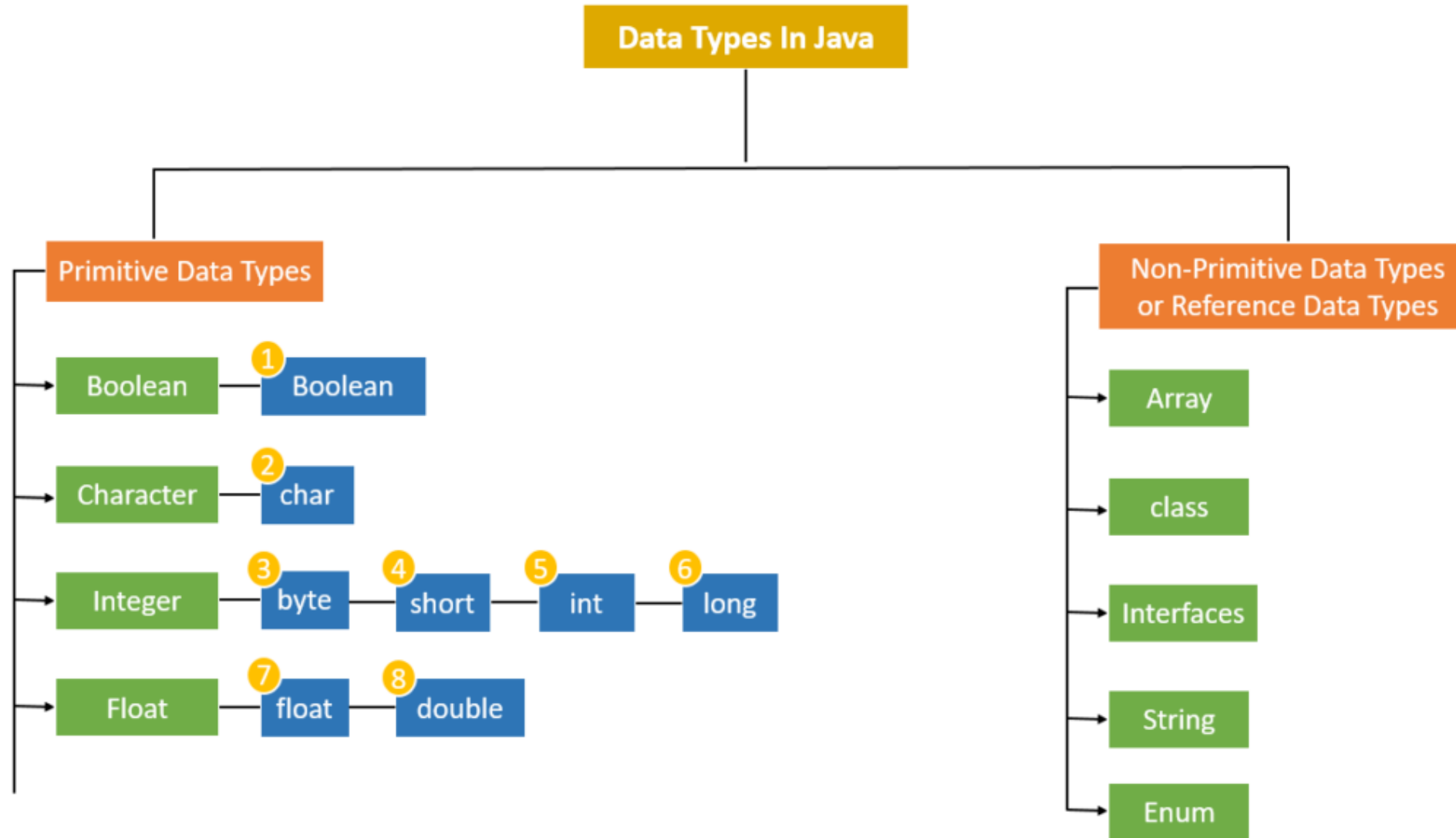
Veri Tipleri->KATEKTER DİZİLERİ STRINGLR

- Stringler karakterlerin(char) birleşimi ile oluşur.
- Veriye direk ulaşamaz.
- Verinin adresine sonrasında kendisine ulaşılır.

0X 05F0C3 ADRESİNE GİT

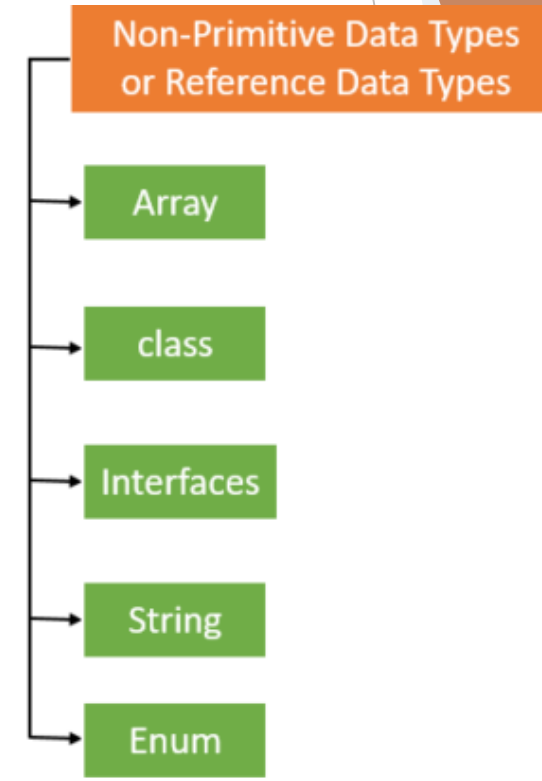
	N
	A
	C
	i
	Y
	E
BİTTİ	\n

JAVA DA VERİ TİPLERİ



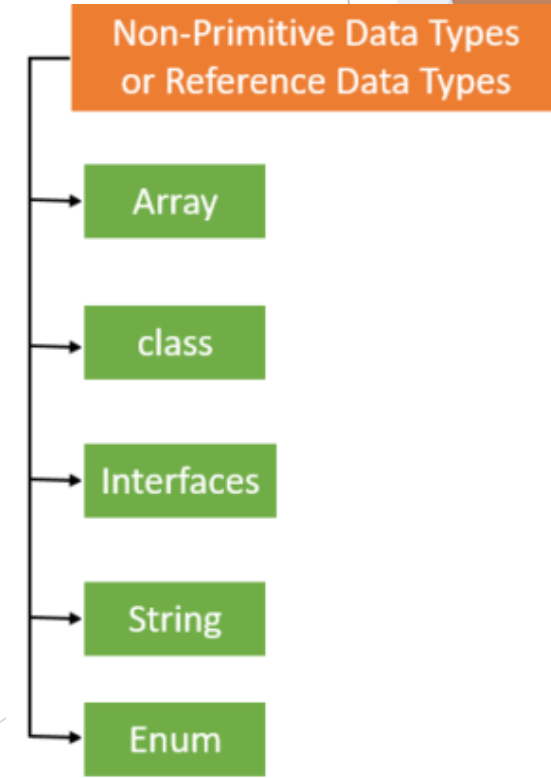
NON Primitive Data Types(İlkel Olmayan Veri Tipleri)

- İlkel olmayan veri tipleri referansları (bellek adresleri) tutar ve bu referanslar üzerinden ilgili nesnelere erişim sağlanır.
- Oluşturulan her class, aynı zamanda bir non-primitive veri tipidir. Bu nedenle non-primitive veri tiplerinin sayısı sınırsızdır.
- Non-primitive veri tiplerinin isimleri her zaman **büyük harfle** başlar.
- Java, non-primitive veri tipleri için boyutlarına göre bellekten farklı miktarlarda yer ayırır.
- bizim atadığımız değerler ile metodlarını içerir.
- java da yazılan kodlarla geliştirilmiştir



NON Primitive Data Types(İlkel Olmayan Veri Tipleri)

- İlkel olmayan veri tipleri referansları (bellek adresleri) tutar ve bu referanslar üzerinden ilgili nesnelere erişim sağlanır.
- Array: Aynı veya farklı veri tiplerinden oluşan diziler.
- Class: Nesne yönelimli programlama sınıflarını temsil eder.
- Interface: Arayüz tanımlamaları.
- Enum: Özel türler veya sabit değerler için kullanılır.
- Object: Tüm nesnelerin ata sınıfı.
- String: Metin dizilerini temsil eder.



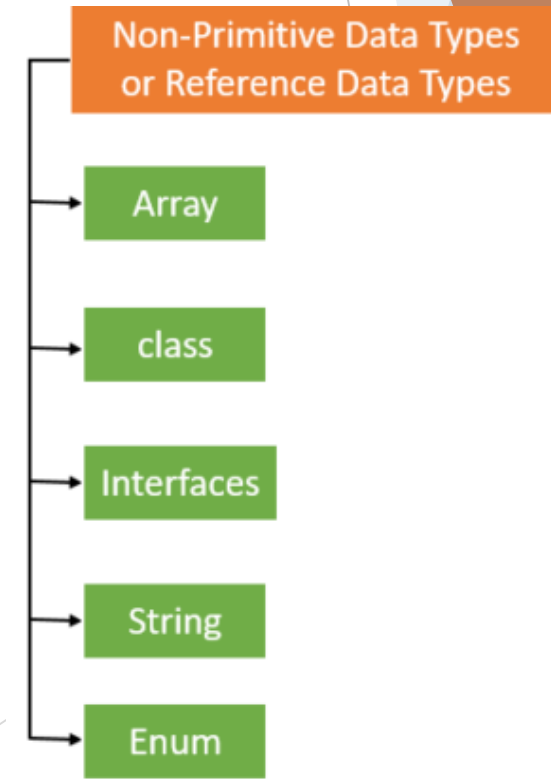
NON Primitive Data Types(İlkel Olmayan Veri Tipleri)

- Veriye direk ulaşamaz.
- Verinin adresine sonrasında kendisine ulaşılır.

0X 05F0C3 ADRESİNE GİT

N
A
C
i
Y
E
\n

BITTİ



Primitive ve Non Primitive Data Types FARKLARI

Özellik	Primitive Data Type	Non-Primitive Data Type
İçerik	Sadece atanan değeri saklar	Hem değeri hem de methodları barındırır
İsimlendirme	Her zaman küçük harfle başlar	Her zaman büyük harfle başlar
Oluşturulma	Java tarafından tanımlanmıştır, 8 tanedir	Hem Java hem de programcı tarafından oluşturulabilir, sınırsızdır
Bellek Kullanımı	Türüne göre sabit boyutta bellek ayırır	Boyutuna bağlı olarak değişken miktarda bellek ayırır

DEĞİŞKENLER

Değişken Nedir?

- Veri saklamak
- Verileri kullanmak
- Veriler üzerinde değişiklik yapmak

Tanımlanan anahtar isimlerdir.

Değişken Tanımlama

```
Veri_Tipi Değişken_adı;  
int      a;
```

Atama Operatörü =

SOLDAKİ DEĞİŞKENE = SAĞDAKİ VERİ
ATANIR

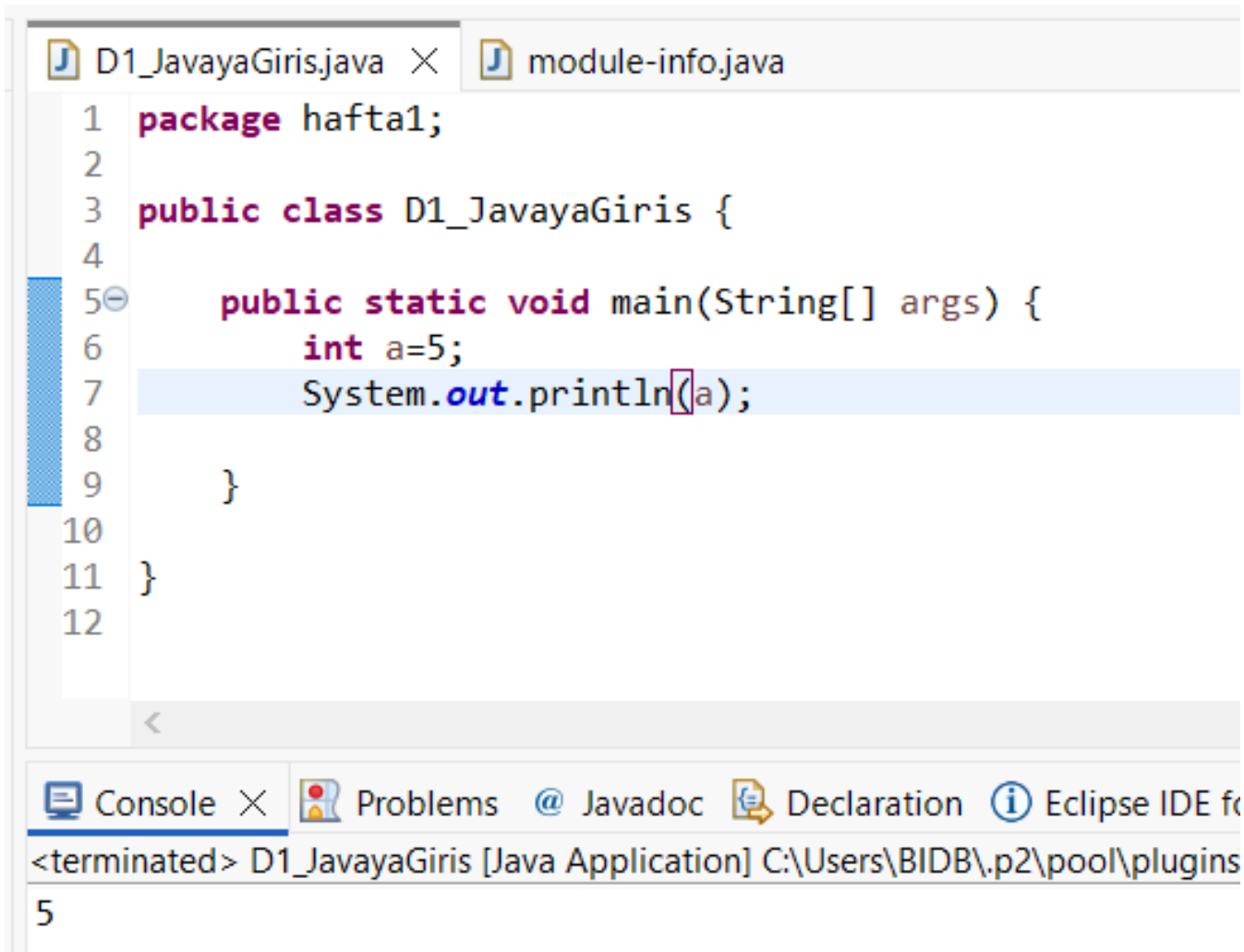
a=5;

Başlangıç Değeri İle Değişken Tanımlama

Veri_Tipi Değişken_adı;

int a =5;

Değişken Tanımlama



```
1 package hafta1;
2
3 public class D1_JavayaGiris {
4
5     public static void main(String[] args) {
6         int a=5;
7         System.out.println(a);
8
9     }
10
11 }
12
```

Console × Problems @ Javadoc Declaration ⓘ Eclipse IDE for Java Developers

<terminated> D1_JavayaGiris [Java Application] C:\Users\BIDB\p2\pool\plugins

5

Değişken Tanımlama Kuralları

- ▶ Değişken adları en fazla 32 karakterden oluşabilir. Geriye kalan karakterler işleme tabi tutulmaz.

int muvaffakiyetsizleştiricileştiriveremeyebileceklerimizdenmissinizcesine = 5;



Değişken Tanımlama Kuralları

- ▶ Değişken adları İngiliz alfabesinde bulunan karakterler (A-Z) veya (a-z) yada rakamlar (0-9) ile yazılmalıdır.
- ▶ Türkçe karakterler, özel karakter veya boşluk karakteri kullanılamaz.
(ğ, Ğ, ş, Ş, ı, İ)

`int deger = 5;`
`int deger1 = 5;`



`int değer = 5;`



Değişken Tanımlama Kuralları

- ▶ Değişken adları herhangi bir rakam ile başlayamaz. İlk karakter bir harf olmalıdır. Sonrakiler rakamlardan oluşabilir.

`int deger = 5;`
`int deger1 = 5;`



`int 1deger = 5;`



Değişken Tanımlama Kuralları

- Aşağıda verilen kelimeler ANSI C 'nin anahtar kelimeleridir (key words) ve değişken ismi olarak kullanılamaz.

Kategori	Anahtar Kelimeler
Kontrol ve Akış	if, else, switch, case, default, for, while, do, break, continue, return
Sınıf ve Nesne	class, interface, extends, implements, new, this, super
Erişim ve Diğer Modifier'lar	public, private, protected, static, final, abstract, synchronized, volatile, transient, strictfp
Temel Veri Tipleri ve Tipler	byte, short, int, long, float, double, char, boolean, void, enum
Hata ve İstisna Yönetimi	try, catch, finally, throw, throws
Paket ve İç Aktarma	package, import
Mantıksal ve Diğer Önemli	instanceof, null, true, false
Rezerved / Kullanılmayan (Geçmişten)	const, goto
Native ve Sistemsel	native

`int while = 5;`



Değişken Tanımlama Kuralları

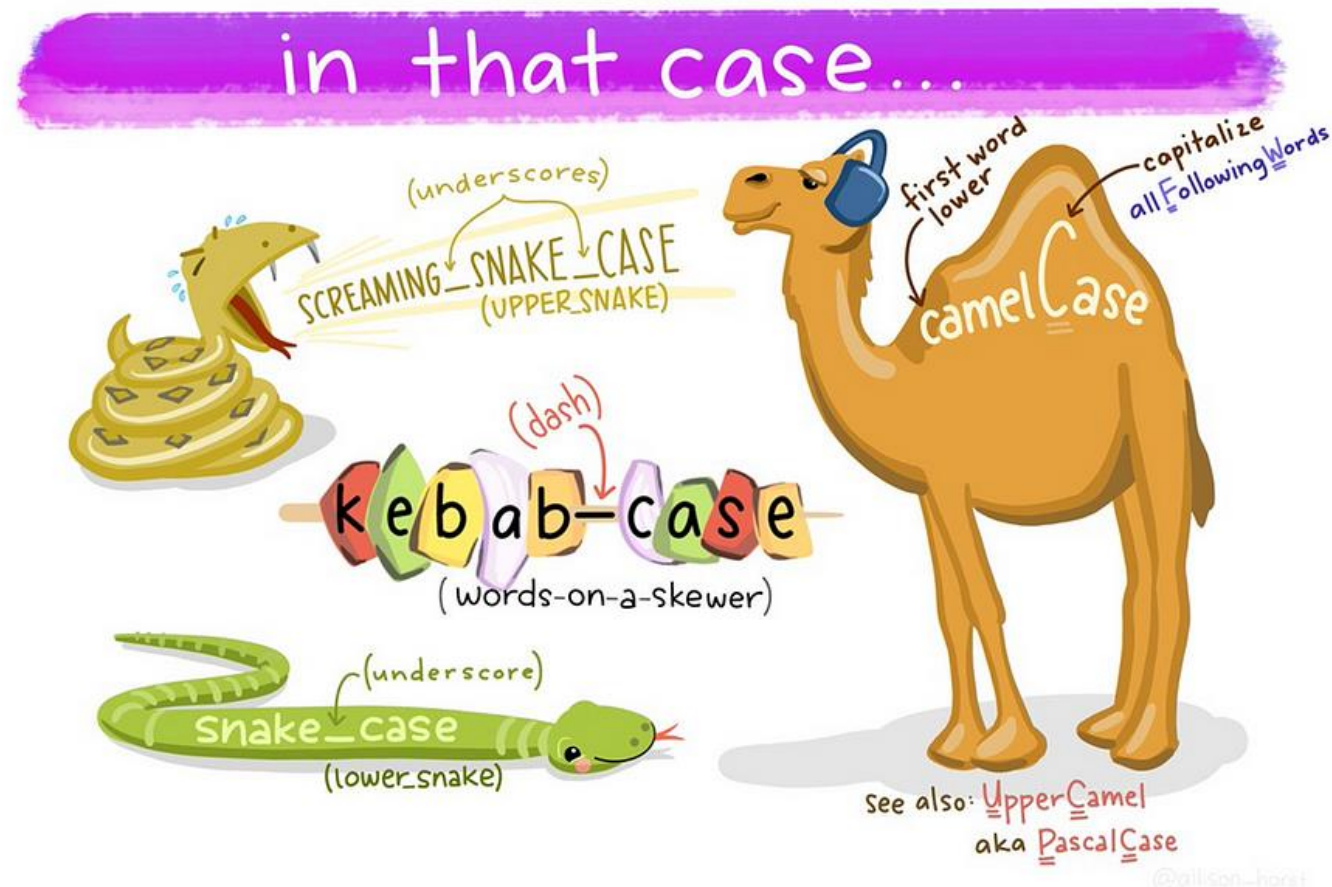
- Programlamada değişken tanımlarken geliştiriciler farklı stilde değişken tanımlar. Bu stillerin ismi vardır. Popüler olanları şunlardır;

`snake_case`

`camelCase`

`PascalCase`

`kebab-case`



Değişken Tanımlama Kuralları

- ▶ **SNAKE_CASE**
- ▶ Snake case kullanımında tüm harfler küçüktür ve değişken tanımlanırken alt tire (_) kullanılır.
- ▶ Upper Snake Case'de kullanılabilir

```
int snake_case=5;  
int SNAKE_CASE=5;
```



Değişken Tanımlama Kuralları

- **Camel case** kullanımında ilk kelime küçük harflerle yazılırken ikinci kelimenin ilk harfi büyük harf ile başlar.

```
int camelCase=5;
```



Değişken Tanımlama Kuralları

- **Pascal Case** Camel case kendi içinde Pascal Case (Upper Camel) şeklinde de kullanılabilir

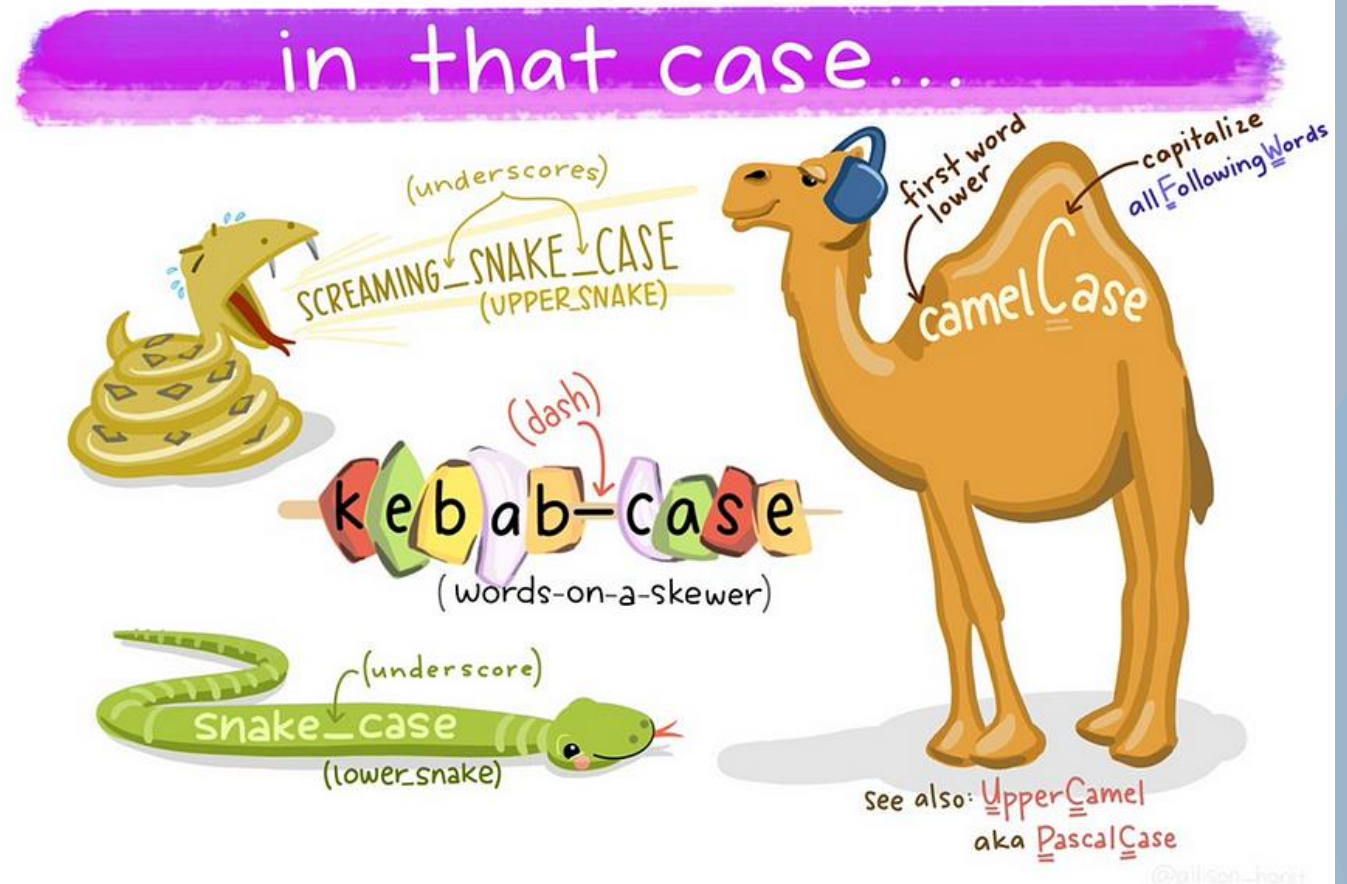
```
int PascalCase=5;
```



Değişken Tanımlama Kuralları

- **Kebab Case** Kebab case, her harf küçük karakterlidir. Değişken isimlendirirken kelimeler arasında tire (-) işareti kullanılır. Snake case farkı alt tire (_) kullanılmamasıdır.

```
int kebab-case=5;
```



Değişken Tanımlama Kuralları

Öğe	İsimlendirme Stili (Case)	Örnek	Notlar
Sınıf (Class)	PascalCase	CustomerAccount, InvoiceManager	Java'da sınıf isimleri her zaman büyük harfle başlar.
Metod (Method)	camelCase	calculateTotal(), getUsername()	Küçük harfle başlar, sonraki kelimeler büyük harfle yazılır.
Değişken (Variable)	camelCase	userName, totalPrice	Metod içi veya sınıf içi değişkenler için kullanılır.
Sabit (Constant)	UPPER_SNAKE_CASE	MAX_SIZE, DEFAULT_TIMEOUT	final static değişkenler için kullanılır.
Paket (Package)	lower.case.with.dots	com.company.project	Küçük harflerle ve nokta ile ayrılır.
Arayüz (Interface)	PascalCase	UserRepository, Serializable	Sınıflar gibi büyük harfle başlar.
Enum	PascalCase	UserStatus, PaymentType	Sınıf isimlendirmesi gibi, sabitleri genelde UPPER_SNAKE_CASE.

💡 Ek Notlar:

- Java'da **kebab-case** kullanılmaz; değişken veya sınıf isimlerinde geçerli değildir.
- camelCase**, **PascalCase**, ve **UPPER_SNAKE_CASE** Java'da en yaygın isimlendirme stilleridir.
- Kod okunabilirliği ve Java standartlarına uyum açısından bu kurallara dikkat etmek önemlidir.

Değişken Tanımlama Kuralları

*D2_DegiskenTanımlamaKurallari.java X

```
1 // Paket isimlendirmesi: küçük harflerle ve noktalar ile ayrılır
2 // Örnek: package com.sirket.proje;
3 package hafta1;
4 //Sınıf isimlendirmesi: PascalCase
5 public class D2_DegiskenTanımlamaKurallari {
6
7     // Sabitler: UPPER_SNAKE_CASE
8     public static final int MAKS_BAKIYE = 10000; // Maksimum bakiye
9     public static final int DEFAULT_ZAMAN_ASIMI = 30; // Dakika cinsinden
10
11     // Değişkenler: camelCase
12     private String kullaniciAdi; // Kullanıcı adı
13     private double bakiye; // Hesap bakiyesi
14
15     // Constructor (yapıcı metodlar da PascalCase)
16     public D2_DegiskenTanımlamaKurallari(String kullaniciAdi, double bakiye) {
17         this.kullaniciAdi = kullaniciAdi;
18         this.bakiye = bakiye;
19     }
20
```

Değişken Tanımlama Kuralları

*D2_DegiskenTanimlamaKurallari.java

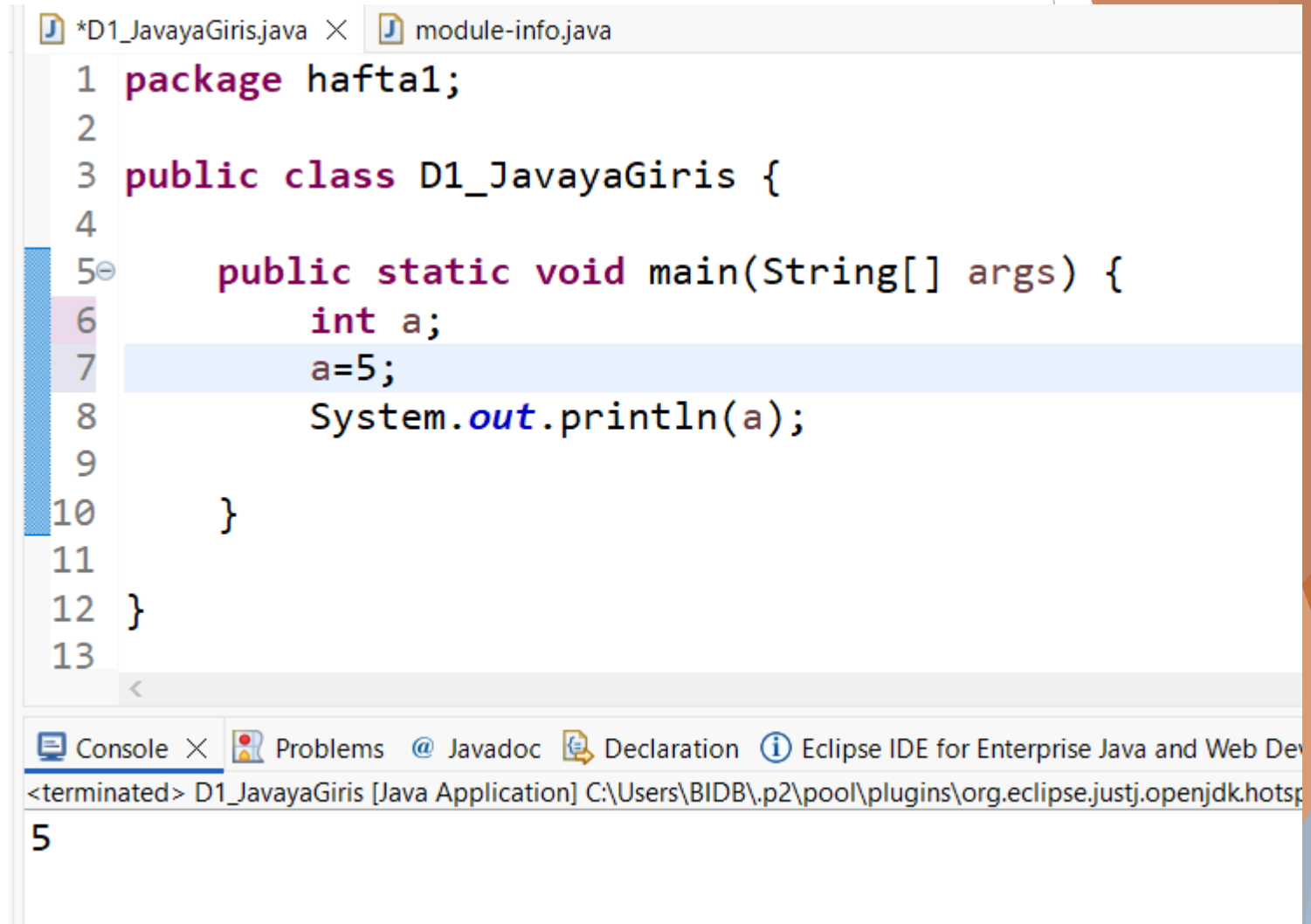
```
20
21 // Metod isimlendirmesi: camelCase
22 public void bakiyeGoster() {
23     // Metod içi değişkenler: camelCase
24     double geciciBakiye = this.bakiye; // Hesap bakiyesini geçici değişkene atama
25     System.out.println("Kullanıcı: " + kullanıcıAdi + " - Bakiye: " + geciciBakiye);
26 }
27
28 public boolean paraCek(double miktar) {
29     if (miktar <= bakiye) {
30         bakiye -= miktar;
31         return true;
32     } else {
33         System.out.println("Yetersiz bakiye!");
34         return false;
35     }
36 }
37 }
38
```


Değişken Tanımlama Kuralları

```
38
39 // Arayüz örneği: PascalCase
40 interface OdemeYapilabilir {
41     // Metodlar camelCase
42     void odemeYap(double miktar);
43 }
44
45 // Enum örneği: PascalCase
46 enum KullaniciDurumu {
47     AKTIF,           // Sabitler: UPPER_SNAKE_CASE
48     PASIF,
49     ENGELLENMIS
50 }
```

Değişken Tanımlama

- ▶ Değişkeni içi boş tanımlayıp
- ▶ Sonra içerisine = atama operatörü ile veri atanabilir.



```
1 package hafta1;
2
3 public class D1_JavayaGiris {
4
5     public static void main(String[] args) {
6         int a;
7         a=5;
8         System.out.println(a);
9
10    }
11
12 }
13
```

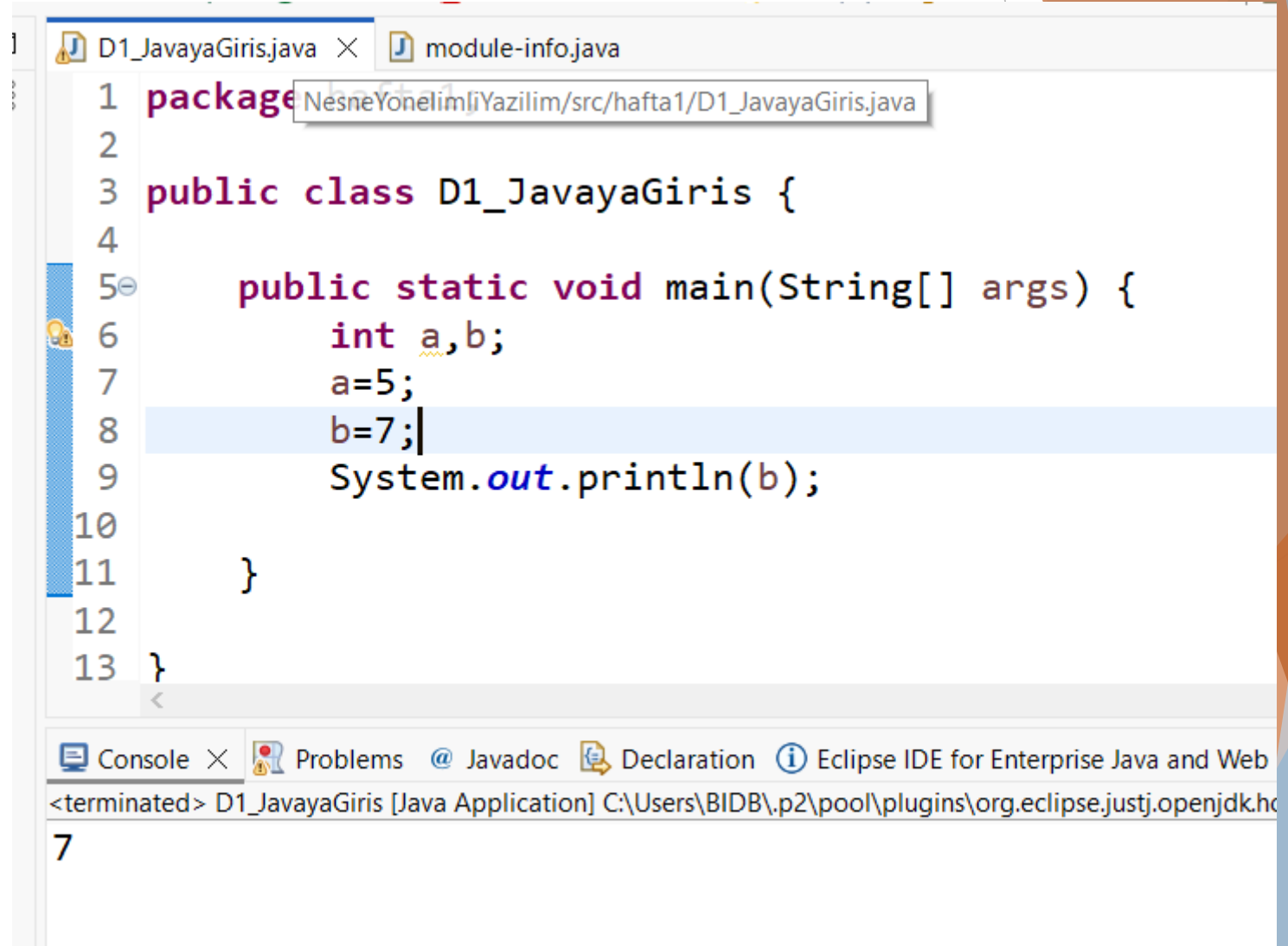
Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Dev

<terminated> D1_JavayaGiris [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk.hotsp

5

Değişken Tanımlama

- Birden fazla aynı veri tipi değişkeni aynı satırda içi boş tanımlayıp
- Sonra içerisine = atama operatörü ile veri atanabilir.



```
1 package NesneYonelimliYazilim/src/hafta1/D1_JavayaGiris.java
2
3 public class D1_JavayaGiris {
4
5     public static void main(String[] args) {
6         int a,b;
7         a=5;
8         b=7;
9         System.out.println(b);
10
11     }
12
13 }
```

Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web
<terminated> D1_JavayaGiris [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hc
7

Başlangıç Değeri İle Değişken Tanımlama

Veri_Tipi **Değişken_adı**;

int **a =5;**

Veri Tipi (Data Type)	Değişken Adı (Variable Name)	=	Örnek Değer (Value)
byte	yas	=	25
short	kisaSayi	=	32000
int	sayi	=	150000
long	nufus	=	9000000000L
float	sicaklik	=	36.5f
double	piSayisi	=	3.14159
char	harf	=	'A'
boolean	aktifMi	=	true
String	isim	=	"Ahmet"

Değişken Tanımlama

```
D4_ToplamaOperatoru.java  D5_DegiskenTanımlama2.java X
4
5- public static void main(String[] args) {
6     // byte 0 den + 255'e tamsayı değerleri için kullanılır
7     byte ogrenciYas=15;
8     System.out.println(ogrenciYas);
9     // short -32768 den + 32767'e tamsayı değerleri için kullanılır
10    short siteNufusu=1200;
11    System.out.println(siteNufusu);
12    // int -2,147,483,648 ile 2,147,483,647'e
13    int sehirNufusu=84234657;
14    System.out.println(sehirNufusu);
15    // long -9,223,372,036,854,775,808 ile 9,223,372,036,854,775,807
16    //Bir sayıyı long tipinde tanımlarken sonuna
17    //mutlaka "L" harfi eklemelisiniz,
18    //aksi halde "Integer number too large"
19    //(Tam sayı çok büyük) hatası alırsınız.
20    long hucreSayisi=123456789012L;
21    System.out.println(hucreSayisi);

```

Console X Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Developers 2025-09 Release

```
<terminated> D5_DegiskenTanımlama2 [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0
15
1200
84234657
123456789012
```

Değişken Tanımlama

```
D4_ToplamaOperatoru.java  *D5_DegiskenTanımlama2.java X
21      System.out.println("İlk sayı 151");
22
23
24  /*float ondalıklı sayılar için
25  -> Java ondalıklı sayıları otomatik olarak double kabul eder.
26  Bu yüzden varsayılan (default) tip double'dır.
27  Eğer float olmasını istiyorsanız, sayının sonuna "F" veya "f" koymalısınız.*/
28  float tisortFiyat = 25.99f;
29      System.out.println("Fiyat: "+ tisortFiyat);
30  double cellWeight = 0.000000000000000000028;
31      System.out.println(cellWeight);
32      //2.8E-19 ==> 2.8 çarpı 10 üzeri -19 demektir
33  }
34
```

Console X Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Developers 2025-09 Release

<terminated> D5_DegiskenTanımlama2 [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe

```
15
1200
84234657
123456789012
Fiyat: 25.99
2.8E-19
```

Değişken Tanımlama

```
D4_ToplamaOperatoru.java  D5_DegiskenTanimlama2.java  *D6_Karakter_StringTanimlama.java X
1 package hafta1;
2
3 public class D6_Karakter_StringTanimlama {
4 public static void main(String[] args) {
5     char isminIlkHarfi = 'N';
6     System.out.println(isminIlkHarfi);
7     System.out.println(isminIlkHarfi+3);
8     //char lar aslında sayısal verilerdir ve + ile toplama işlemi gibi çalışır.
9     String ismiSoyismi="NACİYE MACİT SEZİKLİ";
10    System.out.println(ismiSoyismi);
11 }
12 }
13
```

Console X Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Developers 2025-09 Release

<terminated> D6_Karakter_StringTanimlama [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502

N

81

NACİYE MACİT SEZİKLİ

Değişken Tanımlama

D7_BooleanTanimlama.java ×

```
1 package hafta1;
2 public class D7_BooleanTanimlama {
3
4     public static void main(String[] args) {
5         //Boolean Sadece true (dogru) ve false (yanlış) olmak üzere 2 farklı değer alırlar.
6         boolean durum=false;
7         System.out.println(durum);
8     }
9 }
10
```

Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Developers 2025-09 Release

<terminated> D7_BooleanTanimlama [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (21 Ey
false

+ Birleştirme Operatörü

- İki ifadeyi yan yana gelecek şekilde birleştirir.

```
public static void main(String[] args) {  
    // TODO code application logic here  
    System.out.println("bu benim ilk java projem :) ");  
    System.out.print("Merhaba");  
    System.out.print("Java");  
    System.out.println("Dünyası"+"Nasılsın");  
}
```

```
run:  
bu benim ilk java projem :)  
MerhabaJavaDünyasıNasılsın  
BUILD SUCCESSFUL (total time: 0 seconds)
```

+ Birleştirme Operatörü

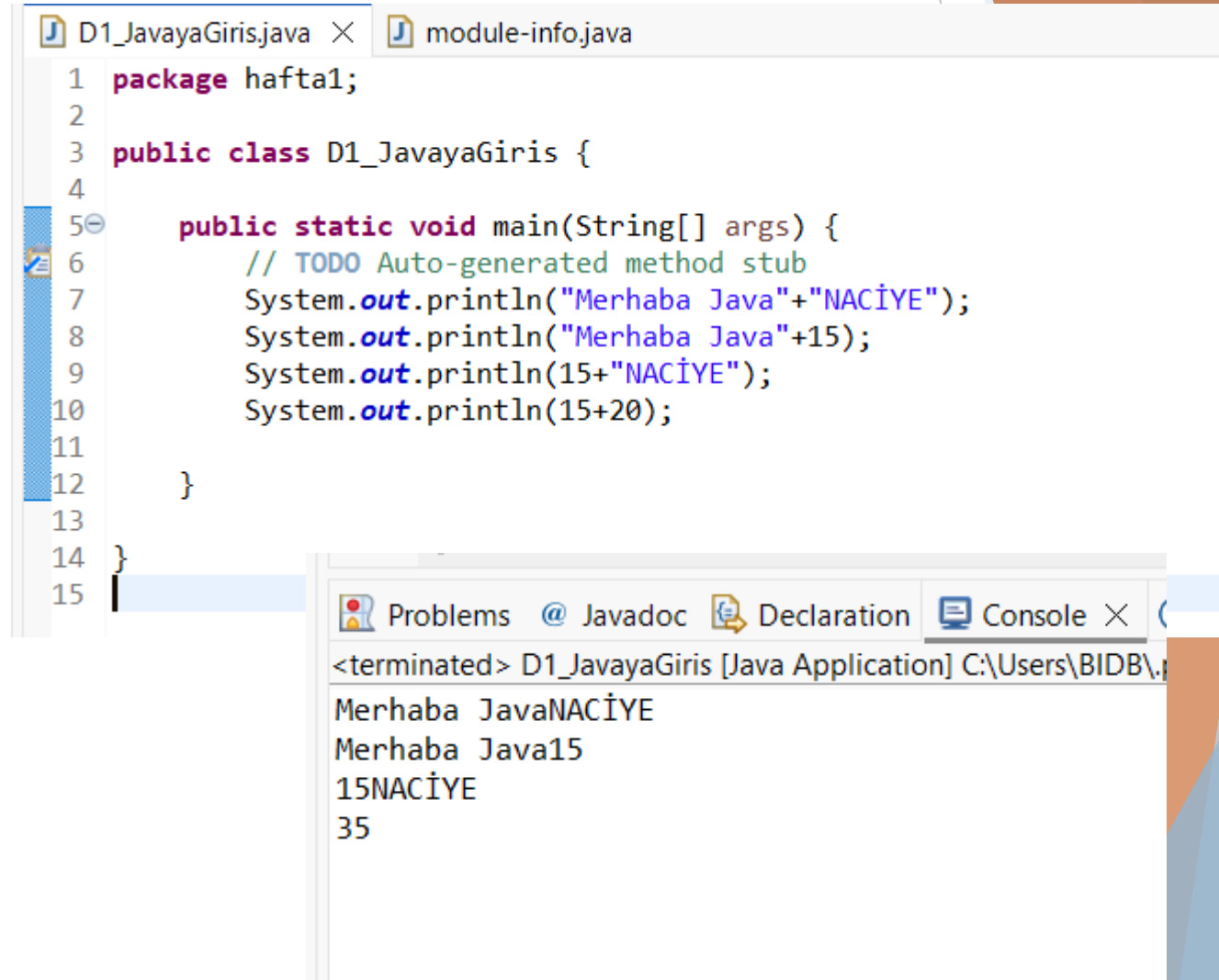
- Metin yanına sayı gelirse onu da metin ile birleştirir.

```
public static void main(String[] args) {  
    // TODO code application logic here  
    System.out.println("bu benim ilk java projem :) ");  
    System.out.print("Merhaba");  
    System.out.print("Java");  
    System.out.println("Dünyası"+15);  
}
```

```
run:  
bu benim ilk java projem :)  
MerhabaJavaDünyası15  
BUILD SUCCESSFUL (total time: 0 seconds)
```

+ TOPLAMA Operatörü

- İki sayı arasına gelirse artık bir toplama operatörüdür.



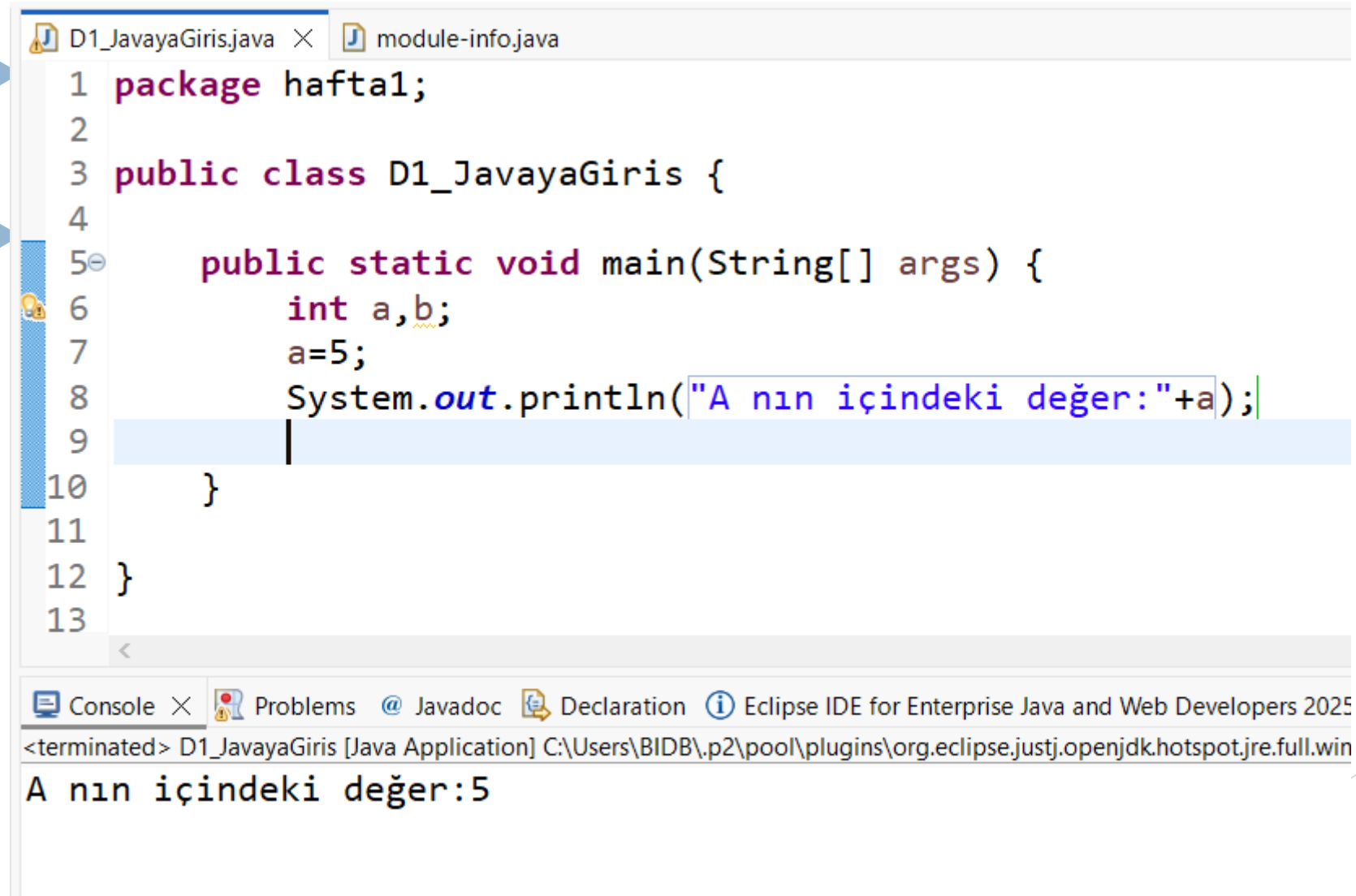
The screenshot shows an IDE with two tabs: `D1_JavayaGiris.java` and `module-info.java`. The `D1_JavayaGiris.java` tab is active, displaying the following code:

```
1 package hafta1;
2
3 public class D1_JavayaGiris {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("Merhaba Java"+"NACİYE");
8         System.out.println("Merhaba Java"+15);
9         System.out.println(15+"NACİYE");
10        System.out.println(15+20);
11    }
12 }
13
14 }
15
```

Below the code editor, the `Console` tab is active, showing the output of the program:

```
<terminated> D1_JavayaGiris [Java Application] C:\Users\BIDB\...
Merhaba JavaNACİYE
Merhaba Java15
15NACİYE
35
```

Birleştirme Operatörü



```
1 package hafta1;
2
3 public class D1_JavayaGiris {
4
5     public static void main(String[] args) {
6         int a,b;
7         a=5;
8         System.out.println("A nın içindeki değer:"+a);
9
10    }
11
12 }
13
```

<terminated> D1_JavayaGiris [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win

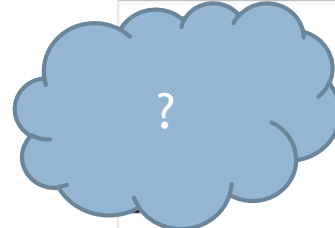
A nın içindeki değer:5

Birleştirme Operatörü

- + nin önünde metin varsa sonraki + lar birleştirme operatörü olur.

```
D1_JavayaGiris.java × module-info.java
1 package hafta1;
2
3 public class D1_JavayaGiris {
4
5     public static void main(String[] args) {
6         int a,b;
7         a=5;
8         b=7;
9         System.out.println("10"+a);
10        System.out.println("10"+a+b);
11        System.out.println(a+b+"10");
12        System.out.println(a+"10");
13
14    }
15
16 }
```

Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java an
<terminated> D1_JavayaGiris [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.op



Toplama Operatörü

```
D4_ToplamaOperatoru.java ×
1 package hafta1;
2
3 public class D4_ToplamaOperatoru {
4
5     public static void main(String[] args) {
6         ///  
iki yanına sayı gelirse toplama operatörü olarak çalışır
7         int a=4,b=98;
8         int toplam=a+b;
9         System.out.println("Sonuç:"+toplam);
10
11     }
12
13 }
14
```

Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Developers 2025-09 Release

<terminated> D4_ToplamaOperatoru [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0

Sonuç:102

Toplama Operatörü

```
*D4_ToplamaOperatoru.java ×
1 package hafta1;
2
3 public class D4_ToplamaOperatoru {
4
5     public static void main(String[] args) {
6         //+ iki yanına sayı gelirse toplama operatörü olarak çalışır
7         int a=4,b=98;
8         int toplam=a+b;
9         System.out.println("Sonuç:"+toplam);
10        System.out.println("Sonuç:"+a+b); //birleştirme operatörü
11        System.out.println("Sonuç:"+a+b)); //toplama operatörü
12        //() işlem önceliği var
13    }
14
```

Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Developers 2025-09 Release

<terminated> D4_ToplamaOperatoru [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v202505

Sonuç:102

Sonuç:498

Sonuç:102

Satır Atlama

```
D8_PrintlnPrintFarki.java ×
1 package hafta1;
2
3 public class D8_PrintlnPrintFarki {
4
5     public static void main(String[] args) {
6         //println satır atlayarak yazar
7         System.out.println("Naciye");
8         System.out.println("Macit");
9         //print satır atlamadan devam eder.
10        System.out.print("Naciye");
11        System.out.print("Macit");
12    }
13 }
```

Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Develop

<terminated> D8_PrintlnPrintFarki [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk.hotsp

Naciye
Macit
NaciyeMacit

OPERATÖRLER

Arımetıksel

Mantıksal

Karşılařtırma

Aritmatiksel Operatörler

Operatör	Açıklama	Örnek	Anlamı
+	toplama	$x + y$	x ve y nin toplamı
-	çıkarma	$x - y$	x ve y nin farkı
*	carpma	$x * y$	x ve y nin çarpımı
/	bölme	x / y	x ve y nin oranı
%	artık bölme	$x \% y$	x / y den kalan sayı

Aritmatiksel Operatörler

➡ **TOPLAMA** ➡ + 5+7

➡ **BİRLEŞTİRME** ➡ + "5"+"7"

```
//TOPLAMA
int a=5,b=4;
System.out.print("Toplama Operatörü");
System.out.println(a+b);
//BİRLEŞTİRME
String x="5",y="4";
System.out.print("Birleştirme Operatörü");
System.out.println(x+y);
```

Toplama Operatörü9
Birleştirme Operatörü54

Aritmatiksel Operatörler

➡ ÇIKARMA ➡

➡ -

➡ 5-4

```
4
5 public static void main(String[] args) {
6     //TOPLAMA
7     int a=5,b=4;
8     System.out.print("Toplama Operatörü");
9     System.out.println(a+b);
10    //BİRLEŞTİRME
11    String x="5",y="4";
12    System.out.print("Birleştirme Operatörü");
13    System.out.println(x+y);
14    //ÇIKARMA
15    System.out.print("Çıkarma Operatörü");
16    System.out.println(a-b);
17    //ÇARPMA
18    System.out.print("Çarpma Operatörü");
19    System.out.println(a*b);
20    //BÖLME
21    System.out.print("Bölme Operatörü");
22    System.out.println(a/b);
23 }
24 }
```

Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web De
<terminated> D9_AritmetikselOperatorler [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.

Toplama Operatörü9
Birleştirme Operatörü54
Çıkarma Operatörü1
Çarpma Operatörü20
Bölme Operatörü1

Aritmatiksel Operatörler

➡ ÇARPMA ➡

➡ *

➡ 5*4

```
4
5 public static void main(String[] args) {
6     //TOPLAMA
7     int a=5,b=4;
8     System.out.print("Toplama Operatörü");
9     System.out.println(a+b);
10    //BİRLEŞTİRME
11    String x="5",y="4";
12    System.out.print("Birleştirme Operatörü");
13    System.out.println(x+y);
14    //ÇIKARMA
15    System.out.print("Çıkarma Operatörü");
16    System.out.println(a-b);
17    //ÇARPMA
18    System.out.print("Çarpma Operatörü");
19    System.out.println(a*b);
20    //BÖLME
21    System.out.print("Bölme Operatörü");
22    System.out.println(a/b);
23 }
24 }
```

Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web De

<terminated> D9_AritmetikselOperatorler [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj

```
Toplama Operatörü9
Birleştirme Operatörü54
Çıkarma Operatörü1
Çarpma Operatörü20
Bölme Operatörü1
```

Aritmatiksel Operatörler

➡ **BÖLME** ➡

➡ **/**

➡ **7/5**

- `int/int =int`
- `int` sayının `integer` sayıya bölüm sonucu yine `integer`dir.

```
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

public static void main(String[] args) {
    //TOPLAMA
    int a=5,b=4;
    System.out.print("Toplama Operatörü");
    System.out.println(a+b);
    //BİRLEŞTİRME
    String x="5",y="4";
    System.out.print("Birleştirme Operatörü");
    System.out.println(x+y);
    //ÇIKARMA
    System.out.print("Çıkarma Operatörü");
    System.out.println(a-b);
    //ÇARPMA
    System.out.print("Çarpma Operatörü");
    System.out.println(a*b);
    //BÖLME
    System.out.print("Bölme Operatörü");
    System.out.println(a/b);
}
```

Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web De
<terminated> D9_AritmetikselOperatorler [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.

Toplama Operatörü9
Birleştirme Operatörü54
Çıkarma Operatörü1
Çarpma Operatörü20
Bölme Operatörü1

Aritmatiksel Operatörler

➡ **KALAN** ➡

➡ **%**

➡ **7%5**

```
D9_AritmetikselOperatorler.java X
5 public static void main(String[] args) {
6     //TOPLAMA
7     int a=5,b=4;
8     System.out.print("Toplama Operatörü");
9     System.out.println(a+b);
10    //BİRLEŞTİRME
11    String x="5",y="4";
12    System.out.print("Birleştirme Operatörü");
13    System.out.println(x+y);
14    //ÇIKARMA
15    System.out.print("Çıkarma Operatörü");
16    System.out.println(a-b);
17    //ÇARPMA
18    System.out.print("Çarpma Operatörü");
19    System.out.println(a*b);
20    //BÖLME
21    System.out.print("Bölme Operatörü");
22    System.out.println(a/b);
23    //KALAN
24    System.out.print("Kalan Operatörü");
25    System.out.println(a%b);
```

Console X Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web

<terminated> D9_AritmetikselOperatorler [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.j

```
Toplama Operatörü9
Birleştirme Operatörü54
Çıkarma Operatörü1
Çarpma Operatörü20
Bölme Operatörü1
Kalan Operatörü1
```

Artırma Operatörleri

➡ Değişkenin içindeki değeri 1 artırır.

```
D10_ArtirmaOperatoru.java X
1 package hafta1;
2
3 public class D10_ArtirmaOperatoru {
4     public static void main(String[] args) {
5         int x=1;
6         int y=x++;
7         //önce atama işlemi gerçekleşti
8         //sonra artırma işlemi gerçekleşti
9         System.out.println("X nin değeri:"+x);
10        System.out.println("Y nin değeri:"+y);
11        int a=1;
12        int b=++a;
13        //önce artırma işlemi gerçekleşti
14        //sonra atama işlemi gerçekleşti
15        System.out.println("A nin değeri:"+a);
16        System.out.println("B nin değeri:"+b);
17    }
18 }
19
```

Console X Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web

<terminated> D10_ArtirmaOperatoru [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.op

```
X nin değeri:2
Y nin değeri:1
A nin değeri:2
B nin değeri:2
```


Azaltma Operatörleri

- ➡ Değişkenin içindeki değeri 1 azaltır.

```
De11AzaltmaOperatoru.java x
1 package hafta1;
2
3 public class De11AzaltmaOperatoru {
4
5     public static void main(String[] args) {
6         int x=1;
7         int y=x--;
8         //önce atama işlemi gerçekleşti
9         //sonra azaltma işlemi gerçekleşti
10        System.out.println("X nin değeri:"+x);
11        System.out.println("Y nin değeri:"+y);
12        int a=1;
13        int b=--a;
14        //önce azaltma işlemi gerçekleşti
15        //sonra atama işlemi gerçekleşti
16        System.out.println("A nin değeri:"+a);
17        System.out.println("B nin değeri:"+b);
18
19    }
20 }
```

Console x Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Development

<terminated> De11AzaltmaOperatoru [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk

```
X nin değeri:0
Y nin değeri:1
A nin değeri:0
B nin değeri:0
```

OPERATÖRLERİN ÖNCELİĞİ

```
De12OperatorlerOnceligi.java ×
1 package hafta1;
2
3 public class De12OperatorlerOnceligi {
4
5     public static void main(String[] args) {
6         int sonuc=(5+7)*3;
7         System.out.println(sonuc);
8         int sonuc1=5+7*3;
9         System.out.println(sonuc1);
10
11     }
12
13 }
```

OPERATÖRLERİN ÖNCELİĞİ

1.Parantez ()

2. Artırma Azaltma ++,--

3.Çarpma, bölme, mod alma *, /, %

4.Toplama ve çıkarma +, -

OPERATÖRLERİN ÖNCELİĞİ

result = 10 + 3 * 5

result = 10 / 2 * 4;

result = 10 - 3 + 2

OPERATÖRLERİN ÖNCELİĞİ

```
6      int sonuc=(5+7)*3;
7      System.out.println("(5 + 7) * 3 =" +sonuc);
8      int sonuc1=5+7*3;
9      System.out.println("5 + 7 * 3 =" +sonuc1);
10     int sonuc2 =10 + 3 * 5;
11     System.out.println("10 + 3 * 5= " +sonuc2);
12     int sonuc3 =10 / 2 * 5;
13     System.out.println("10 / 2 * 5= " +sonuc3);
14     // * ve / için soldan okumaya başlar
15     int sonuc4 =10 - 2 + 5;
16     System.out.println("10 - 2 + 5= " +sonuc4);
17     // - ve + için soldan okumaya başlar
18
```

Console × Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Development

<terminated> De12OperatorlerOnceligi [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.h

```
(5 + 7) * 3 =36
5 + 7 * 3 =26
10 + 3 * 5= 25
10 / 2 * 5= 25
10 - 2 + 5= 13
```

Bileşik Artırma Operatörleri

```
DeBilesikAtamaOperatoru.java X
1 package hafta1;
2
3 public class DeBilesikAtamaOperatoru {
4     public static void main(String[] args) {
5         int x;
6         x=7; //atama operatörü
7         //x=x+3;
8         //x değerinin içerisindeki veriye 3 eklenip yeniden x e atanıyor
9         //bunun yerine işlemi kısaltmak için aşağıdaki bileşik artırma operatörü kullanılabilir.
10        x+=3; //x=x+3; ile aynı işlemi yapar
11        System.out.println(x);
12
13    }
14 }
15
```

Console X Problems @ Javadoc Declaration Eclipse IDE for Enterprise Java and Web Developers 2025-09 Release

<terminated> DeBilesikAtamaOperatoru [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (21 Eyl 2025 2

10

Bileşik Artırma Operatörleri

Operatör	Açıklama	Örnek	Anlamı
=	atama	<code>x = 7;</code>	<code>x = 7;</code>
+=	ekleyerek atama	<code>x += 3</code>	<code>x = x + 3</code>
-=	eksilterek atama	<code>x -= 5</code>	<code>x = x - 5</code>
*=	çarparak atama	<code>x *= 4</code>	<code>x = x * 4</code>
/=	bölerek atama	<code>x /= 2</code>	<code>x = x / 2</code>
%=	bölüp, kalanını atama	<code>x %= 9</code>	<code>x = x % 9</code>
++	bir arttırma	<code>x++</code> veya <code>++x</code>	<code>x = x + 1</code>
--	bir azaltma	<code>x--</code> veya <code>--x</code>	<code>x = x - 1</code>

Mantıksal ifadeler



DOĞRU
TRUE



YANLIŞ
FALSE

Karşılaştırma Operatörler

== Eşit mi?

!= Eşit değil mi?

> Büyük mü?

< Küçük mü?

>= Büyük veya Eşit mi?

<= Küçük veya Eşit mi?

Karşılaştırma Operatörler

== Eşitse

s1=5;

s2=7;

s3=5;

s1==s2



s1==s3



Karşılaştırma Operatörler

!= Eşit Değil mi

s1=5;

s2=7;

s3=5;

s1!=s2



s1!=s3



Karşılaştırma Operatörler

> Büyük mü?

```
s1=5;  
s2=7;  
$s3=4;
```

$s1 > s2$



$s1 > s3$



Karşılaştırma Operatörler

< Küçük mü?

s1=5;

s2=7;

s3=4;

s1<s2



s1<s3



Karşılaştırma Operatörler

>= Büyük veya Eşit mi?

s1=5;

s2=7;

s3=5;

s1>=s2



s1>=s3



s2>=s3



Karşılaştırma Operatörler

<= Küçük veya eşitmi mü?

s1=5;

s2=7;

s3=5;

s1<=s2



s1<=s3



Mantıksal Operatörler

► **VE=> &=> * =>çarpma**

```
int ort=10;  
System.out.println( ort>20 & ort<50 );  
//           false & true  0*1 =0 False
```

```
int ort=80;  
System.out.println( ort>20 & ort<50 );  
//           true  & false  1*0 =0 False
```

```
int ort=30;  
System.out.println( ort>20 & ort<50 );  
//           true  & true   1*1 =1 True
```

	1.Koşul ort>20	2.Koşul ort<50	Sonuç
Durum1	0	0	0
Durum2	0	1	0
Durum3	1	0	0
Durum4	1	1	1

Mantıksal Operatörler

► **VEYA** => **||** => + => **Toplama**

```
int ort=10;  
System.out.println( ort>20 || ort<50 );  
//           false || true  0+1=1 True
```

```
int ort=80;  
System.out.println( ort>20 || ort<50 );  
//           true  || false  1+0=1 True
```

```
int ort=30;  
System.out.println( ort>20 || ort<50 );  
//           true  || true   1+1=1 True
```

	1.Koşul ort>20	2.Koşul ort<50	Sonuç
Durum1	0	0	0+0=0
Durum2	0	1	0+1=1
Durum3	1	0	1+0=1
Durum4	1	1	1+1=1

Mantıksal Operatörler

- ▶ `a = 5`
- ▶ `b = 10`
- ▶ `c = 3`
- ▶ `result = (a < b) & (c > a)`

- ▶ `x = True`
- ▶ `y = False`
- ▶ `z = True`
- ▶ `result = (x || y) & z`