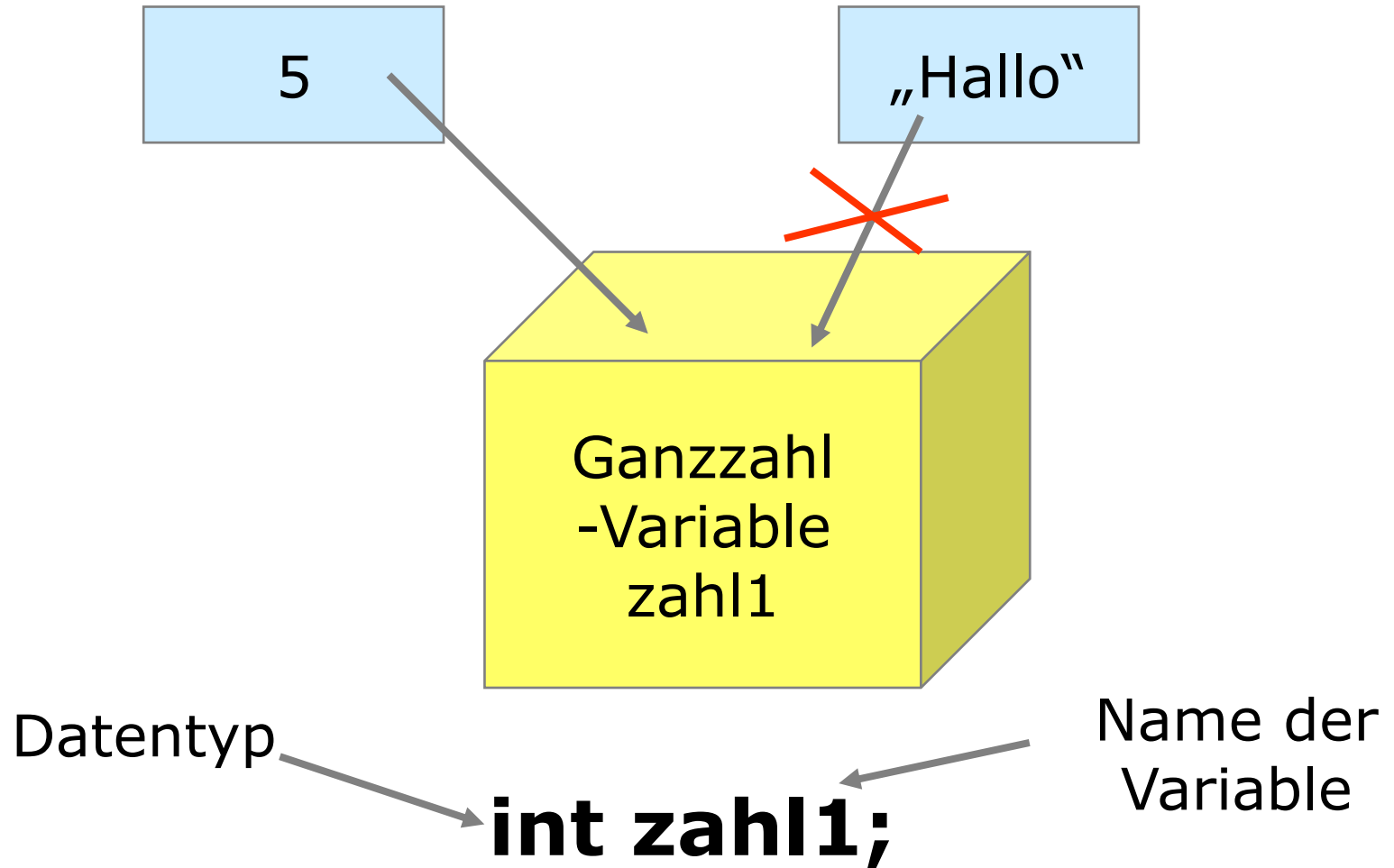




Variablen und Datentypen

Datentypen und Variablen



Konventionen für Variablennamen

- Muss mit einem (**kleinem**) Buchstaben beginnen: ~~2zahl~~
- Mehrere Worte werden groß geschrieben (camelCase):
meineTolleIntVariable
- Präfixe möglich: iZahl1
- Kann beliebig lang sein, aber darf kein Leerzeichen enthalten.
- Schlüsselwörter sind nicht erlaubt: ~~class~~, ~~while~~



Datentypen: Top 4

int

Ganzzahlige Werte

```
int i = 5;
```

double

Kommazahlen

```
double zahl = 3.1;
```

string

Text

```
string text = "Hallo";
```

bool

Wahrheitswerte (true,false)

```
bool ende = false;
```

Datentypen: Wertebereiche und Platzbedarf

Typname	Bereich	Platzbedarf
bool	true, false	1 Bit
char	Unicode-Zeichen von \u0000 ... \uffff	16 Bit
byte	-128 ... 127	8 Bit, mit Vorzeichen
short	-32768 ... 32767	16 Bit, mit Vorzeichen
int	-2147483648 ... 2147483647	32 Bit, mit Vorzeichen
long	-9223372036854775808 ... 9223372036854775807	64 Bit, mit Vorzeichen
float	$\pm 3,40282347E+38$... $\pm 1,40239846E-45$	32 Bit
double	$\pm 1,7976931348623157E+308$... $\pm 4,94065645841246544E-324$	64 Bit

Zusatz Datentypen

Werttypen vs Verweistypen

- **Werttypen**

- `int, long, float, double, char, bool`
- `enum E {...},`
- `struct S {...}`: wie `class`, keine Vererbung, zum Speichern von Daten
- `(T1, T2, ...)` //Tupel

- **Verweistypen**

- `string`
- `object`
- `class c {...}`
- `interface I {...}`: Vertrag über öffentliche Members, kann als Datentyp genutzt werden, simuliert Mehrfachvererbung
- `int[] a, int[,] a2, int[][] aJagged` // arrays
- `delegate int D(...)`: Methoden werden Entitäten, simuliert Pointer auf Methoden

- **Nullable Typen: `int?`, `string?`, ...**

- `int? myNullie=null; myNullie = 1;`

Arbeitsauftrag



**Kreuzworträtsel
in UE02-Datentypen lösen**

Definition und Wertzuweisungen von Variablen

- **Definition**

- `int iZahl1;`

**Darf nur einmal
im Programm
definiert werden.**

- **Definition und Initialwert**

- `int iZahl1=10;`

- **Definition von mehreren Variablen**

- `long lZahl1=600000000000, lZahl2=-7, x;`

- **Konstante**

- **const** `double PI=3.14159265; //Wert kann nicht mehr geändert werden.`

- **Wertzuweisungen**

- `iZahl1=7;`

**Kann beliebig oft
genutzt werden.**

Beispiel: Datentypen und Wertzuweisungen

- Ganzzahlentypen: int, long
 - long lZahl1;
 - lZahl1=100;
- Gleitkommazahlen: float, double
 - float fZahl1=3.5f; // ohne f wird die Zahl als double interpretiert.
 - double x=3.5;
- Logische Variablen: boolean
 - bool ende=false;
- Zeichen: char
 - char zeichen1;
 - zeichen1='c';
- Zeichenketten: string
 - string text = "Hallo";

Arbeitsauftrag



Arbeitsblatt
Wertzuweisungen und Datentypen
Aufgabe 1-4



Operatoren

Operatoren

- Arithmetische Operatoren:
 - `+ - * /`
 - Bsp: `3+5`, `4/i`, ...
 - Modulo-Operator: `%`
 - `int a=10, b=2;`
`int erg=a%b;`
`Console.WriteLine(erg);` -> 0 da `10=2*5+0`
 - Inkrement- und Dekrementoperatoren
 - `int c=10;`
`c++;` // c -> 11 ist das gleiche wie `c=c+1;`
`c--;` // c -> 10
- Mathematische Ausdrücke:
 - `Math.sqrt(9)` // Wurzel, liefert den Wert 3 zurück
 - `Math.sin(0)` // Sinus, liefert den Wert 0 zurück
 - `Math.abs(-3)` // Betrag, liefert den Wert 3 zurück

Programm Notendurchschnitt

- Note SAE=1
- Note ITS=1
- Note BWL=1
- $\text{Durchschnitt} = (\text{SAE} + \text{ITS} + \text{BWL}) / 3$
- Ausgabe: Der Notendurchschnitt beträgt 1.0



Operatoren 2

- Vergleichsoperatoren

- **==** vergleicht zwei Variablen oder Werte

- `float fZahl1 = 3.5f;`
`if (fZahl1 == 3.5f) { fZahl1 = 0f; }`

- **!=** ungleich, `<`, `<=`; `>`, `>=`

- Liefert bool zurück, also true oder false;

- `bool erg;`
`long x=500000;`
`erg = (x==30000); // erg -> false`

- Logische Operatoren

- **&&** (und) **||** (oder) **!** (nicht)

- `bool erg;`
`erg = true && true; // true da beide wahr sind`
`erg = true || false; // true da mindestens eins wahr ist`
`erg = !(true || true) // false, da der Wert der Klammer negiert wird`

- `int x;`
`if (x<15 && x>5) Console.WriteLine(„zwischen 5 und 15“)`
`char zeichen;`
`if (zeichen=='c' || zeichen=='C') Console.WriteLine(„Taste C gedrückt!“);`