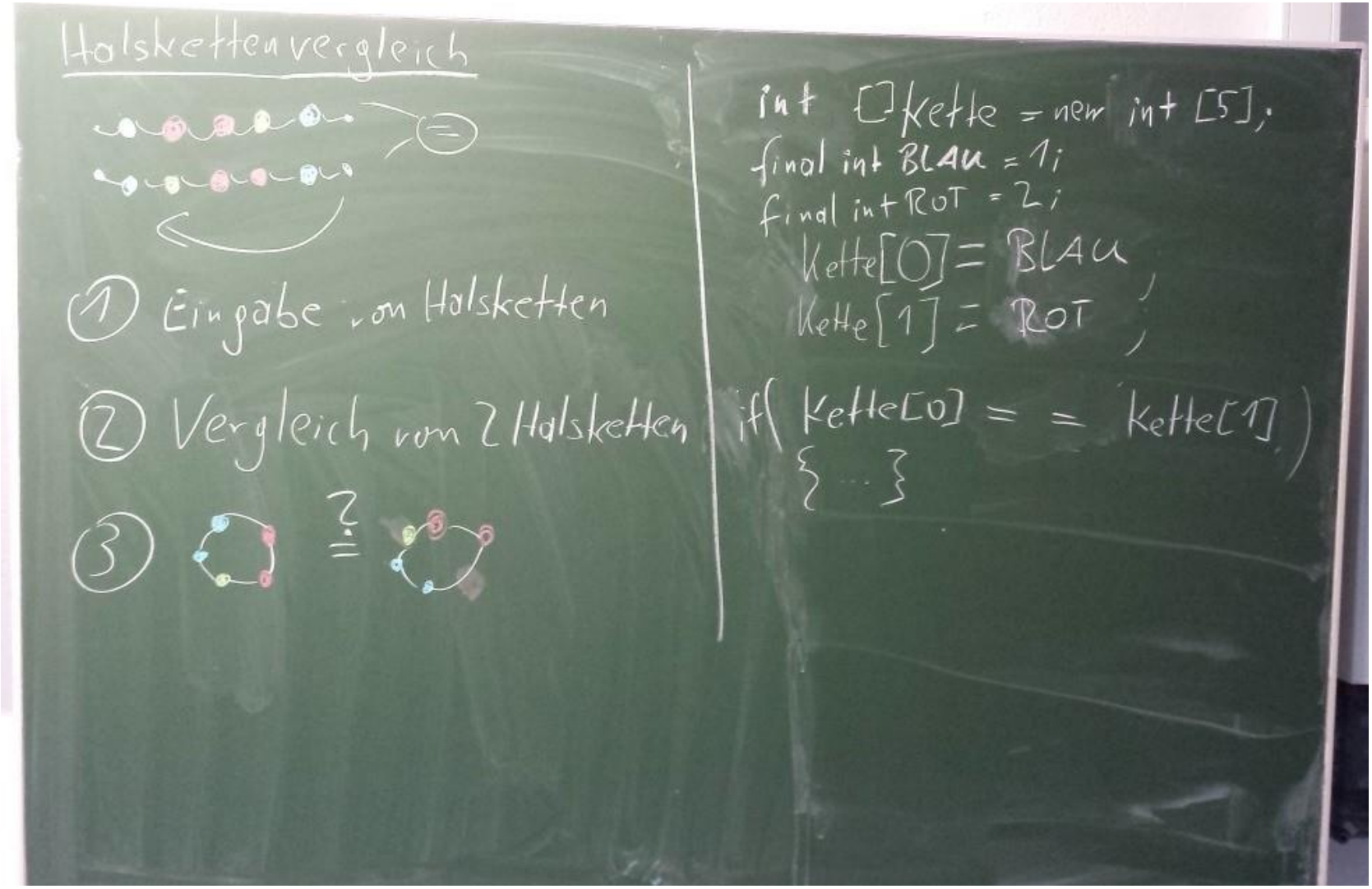
Aufgabe
Halskette vergleichen

Aufgabe


Halskettenvergleich
1 Eingabe von Halsketten
2 Vergleich von 2 Halsketten
3
?
int [] kette = new int [5];
final int BLAU = 1;
final int ROT = 2;
kette[0] = BLAU;
kette[1] = ROT;
if(kette[0] == = kette[1])
{ ... }

Exkurs Konstanten 1/3

- Die Farbe der Perlen lassen sich als Strings darstellen:

  string[] kette1 = {"blau", "rot", "gelb", "rot" };

- Statt dem Speichern von Strings ist es effizienter Zahlen zu speichern:

  int[] kette1 = {0, 1, 2, 1};

- Problem: Welche Zahl entspricht welcher Farbe?
- Ausweg: Konstanten

```
int BLAU=0;
int ROT=1;
int GELB=2;
int[] kette1 = {BLAU, ROT, GELB, ROT};
```

Exkurs Konstanten 2/3

- Damit niemand aus Versehen die Werte der Konstanten verändert, werden sie als unveränderlich deklariert: const

  const int BLAU=0; …

- Die Ausgabe des Arrays ergibt die Zahlen, nicht die Farbe:

  Console.WriteLine(kette1[0]); ↳ 0 // da int[] kette1 = {0, 1, 2, 1};

- Definiere ein Array mit den Farbnamen:

  string[] farbnamen={"Blau", "Rot", "Gelb"};

  Console.WriteLine(farbname[kette1[0]]); ↳ "Blau"

Exkurs Konstanten 3/3

Zusammenfassung

```
const int BLAU=0;
const int ROT=1;
const int GELB=2;
string[] farbnamen={"Blau", "Rot", "Gelb"};
int[] kette1 = {BLAU, ROT, GELB, ROT};
```

Definition Konstanten

```
if (kette1[i]==ROT) {…}
```

Abfrage von Zahlen mit Namen der Konstante

```
for(int i=0;i<kette1.length;i++) {
  CW(farbnamen[kette1[i]]);
}
↳ Blau Rot Gelb Rot
```

Ausgabe der Namen der Konstanten