

Aufgabe4: Radfahrspaß

Lösungsidee:

Teilaufgabe 1:

Für das erste Abfahren des Parcours, um zu überprüfen, ob der Parcours möglich ist, muss sich der BWInf-Fahrer drei ganzzahlige Zähler merken.

Den ersten Zähler benutzt er, um alle ' _ ' zu zählen, d.h. +1 bei einem geraden Streckenabschnitt.

Die zweite Zahl zählt er immer +1, wenn ein ' \ ' als Streckenabschnitt kommt und wenn ein ' / ' als Streckenabschnitt kommt, zieht er -1 ab. Für den Fall, dass diese Zahl unter Null fallen würde durch einen ' / ' Abschnitt, muss er überlegen, ob der erste Zähler noch > 0 ist, wenn ja, muss dem ersten Zähler -1 abgezogen werden, wenn nein, ist der Parcours nicht Regelkonform befahrbar.

Den dritten Zähler benutzt er um zu überprüfen, ob er es schafft, bis zum Ende des Parcours wieder stehen zu bleiben. Er addiert also bei jedem ' \ ' +1 auf diesen Zähler. Da dieser Zähler aber nur benötigt wird, um zu sehen, ob der Fahrer am Ende stehen bleiben kann, wird nur -1 abgezogen, wenn der dritte Zähler > 0 ist und ein ' _ ' oder ' / ' Streckenabschnitt kommt.

Teilaufgabe 2:

Um eine genaue Anweisung auszugeben, wie der Fahrer sich bei jeder Geraden entscheiden sollte, um den Parcours zu schaffen, werden sich bei dem ersten Einlesen der Datei zwei Ganzzahlen gemerkt und beim zweiten Einlesen noch ein Zähler, der sich die Geraden von der Stelle an merkt ab der nicht mehr sicher ist, ob sie + oder - gesetzt werden sollen.

Zu dem erste Zähler wird immer +1 addiert, wenn der zweite Zähler aus Teilaufgabe 1 (+ bei ' / ' und - bei ' \ ') unter 0 fallen würde und eine Gerade "auf beschleunigen gesetzt wird". Mit diesem Zähler weiß man, wie viele Geraden man am Anfang des Parcours auf + setzten muss.

Der zweite Zähler merkt sich die Geschwindigkeit, die der Fahrer am Ende des Parcours hat d.h. alle Geraden die schon sicher auf + gesetzt wurden, werden auf die Geschwindigkeit addiert, dazu addiert werden noch alle ' \ ' Abschnitte und abgezogen werden alle ' / ' Streckenabschnitte.

Der neue Zähler bei dem zweiten Einlesen des Parcours wird immer +1 addiert, wenn eine Gerade ab dem Punkt kommt, wo nicht sicher ist, ob die Gerade + oder - gesetzt werden soll.

Die Geschwindigkeit, die der Fahrer am Ende hat, wird von den noch übrigen Geraden abgezogen. So viele Geraden werden dann abwechselnd auf + und - gesetzt und abschließend werden für alle übrig gebliebenen Geraden noch eine Anweisung zum Bremsen ausgegeben.

Umsetzung:

Die Lösungsidee wird in C# umgesetzt.

Es wird die Eingabedatei eingelesen und drei Integer erstellt. Diese werden genauso hochgezählt wie in der Lösungsidee für Teilaufgabe 1 beschrieben.

Außerdem werden noch zwei weitere Integer erstellt. Diese werden so behandelt wie in der Lösungsidee für Teilaufgabe 2 beschrieben.

Am Ende wird in einem Ausgabeformat:

hintereinander Beschleunigen +'s abwechselnd Beschleunigen/Bremsen +/-'s nur Bremsen -'s

Als Beispiel die Lösung der Testeingabe 4: 1001498+'s 158+/-'s 4-'s.

Hier soll der Fahrer 1001498 mal hintereinander beschleunigen dann soll er 158 abwechselnd beschleunigen und dann bremsen d.h. das er die Hälfte von dem Wert(158) abwechselnd Beschleunigen und Bremsen soll. Zum Schluss muss er noch 4 mal hintereinander bremsen.

Quelltext:

Die Funktion für das erste Einlesen der Datei

```
StreamReader reader = new StreamReader(path);

//jeden char aus der .txt Datei einzeln lesen
char ch;
//die addition der Werte von '/' und '\'
int plusMinus = 0;
int neutral = 0;
//wertet die geraden und schraegen nach oben mit den schraegen nach unten auf
int minus = 0;
do
{
    ch = (char)reader.Read();
    if(DebugMode)
        Console.WriteLine(ch);

    if (ch == '/')
    {
        plusMinus--;
        if (minus > 0)
            minus--;
    }

    else if (ch == '\\')
    {
        plusMinus++;
        minus++;
    }

    else if (ch == '_')
    {
        neutral++;
        if (minus > 0)
            minus--;
    }

    //wenn eine negative Geschw. auftreten wuerde
    if (plusMinus < 0)
    {
        //und es noch geraden gibt
        if (neutral > 0)
        {
            plusMinus++;
            plusses++;
            neutral--;
        }
        else
        {
            Console.WriteLine(notPossible);
            reader.Close();
            return null;
        }
    }
} while (!reader.EndOfStream);
```

Die Funktion für das zweite Einlesen der Datei

```
do
{
    char ch = (char)reader.Read();

    if (ch == '\\')
        geschw++;

    else if (ch == '/')
        geschw--;

    if (ch == '_' && plusses > 0)
    {
        plusses--;
        geschw++;
    }
    else if (ch == '_' && plusses == 0)
    {
        geraden++;
    }
} while (!reader.EndOfStream);
```

Beispiele:

Parcours0: Nicht Möglich

Parcours1: Nicht Möglich

Parcours2: Nicht Möglich

Parcours3: Nicht Möglich

Parcours4: 1001498+'s 158+/-'s 4-'s

Parcours5: Nicht Möglich

Parcours6: Nicht Möglich

Parcours7: Nicht Möglich

Parcours8: Nicht Möglich

Parcours9: 49992889+'s 136+/-'s 184-'s