

Beispiele und Strings

Beispiel: Reisebuchungssystem



Beispiel 1: Reisebuchungssystem

- Ein Reisebuchungssystem erlaubt es, eine Reise für private und geschäftliche Zwecke zu buchen.
- Das System bietet verschiedene Reisen an:
 - Flüge
 - Zugreisen
 - Schiffsausflüge
- Der Preis einer Reise wird aus der Reisedistanz in Kilometern errechnet. Abhängig von der Art der Reise kostet ein Reisekilometer 1,50 € (Flug), 1,20 € (Schiff) oder 0,90 € (Zug).

Beispiel 1: Reisebuchungssystem

- Wenn eine Reise von einer Privatperson gebucht wird, muss die Mehrwertsteuer in Höhe von 19% auf den Reisepreis aufgeschlagen werden.
- Bei einem Geschäftsreisenden wird ein Zuschlag von 0,05 Euro pro KM berechnet.
- Dauert die Reise mehr als 10 Tage, so verringert sich der Reisepreis bei Zugreisen um 24%.

Beispiel 1: Reisebuchungssystem

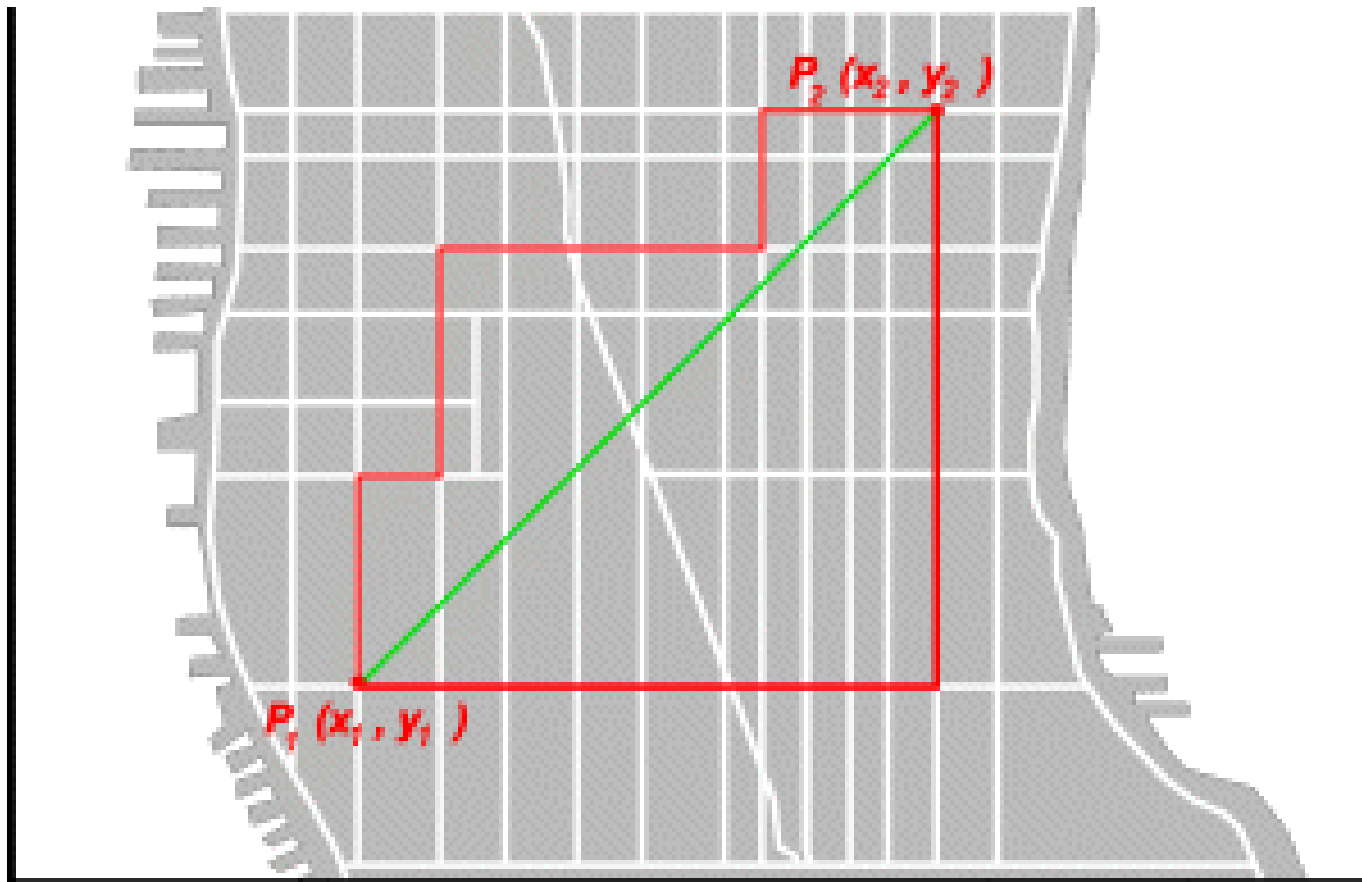
- Ein Reiseziel wird durch 2D-Koordinaten (x, y) in einer vereinfachten Landkarte dargestellt.
- Für Zug und Schiff entspricht die Entfernung zwischen zwei Reisezielen der **Manhattan-Distanz** zwischen den beiden 2D-Koordinatenpunkten.
- Flugzeuge hingegen fliegen auf dem direkten Weg (Luftlinie).

Manhattan Distanz

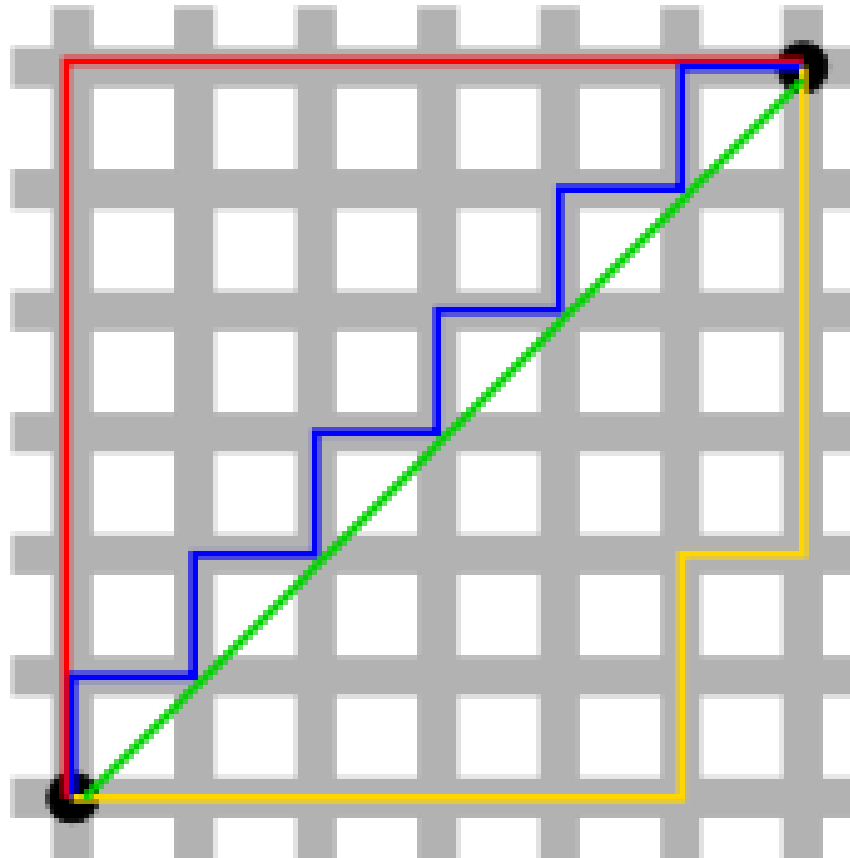


Beispiel 1: Reisebuchungssystem

Manhattan Distanz



Manhattan Distanz



Beispiel 1: Reisebuchungssystem

Das Buchungssystem ist in der Lage,

- eine Schiffs-, Zug- oder Flugreise
- für einen Privat- oder Geschäftsreisenden anzulegen
- und hierfür die Distanz und den Preis der Reise zu berechnen und anzuzeigen

Exkurs: String-Operationen

Beispiel-String: „Hello World“

char	<u>charAt</u> (int index) Returns the char value at the specified index.
boolean	<u>endsWith</u> (<u>String</u> suffix) Tests if this string ends with the specified suffix.

int	<u>indexOf</u> (int ch) Returns the index within this string of the first occurrence of the specified character.
int	<u>indexOf</u> (int ch, int fromIndex) Returns the index within this string of the first occurrence of the specified character, starting the search at the specified index.
int	<u>indexOf</u> (<u>String</u> str) Returns the index within this string of the first occurrence of the specified substring.
int	<u>indexOf</u> (<u>String</u> str, int fromIndex) Returns the index within this string of the first occurrence of the specified substring, starting at the specified index.

int	<u>lastIndexOf</u> (int ch) Returns the index within this string of the last occurrence of the specified character.
int	<u>lastIndexOf</u> (int ch, int fromIndex) Returns the index within this string of the last occurrence of the specified character, searching backward starting at the specified index.
int	<u>lastIndexOf</u> (<u>String</u> str) Returns the index within this string of the rightmost occurrence of the specified substring.
int	<u>lastIndexOf</u> (<u>String</u> str, int fromIndex) Returns the index within this string of the last occurrence of the specified substring, searching backward starting at the specified index.

boolean	<u>isEmpty()</u> Returns true if, and only if, <u>length()</u> is 0.
int	<u>length()</u> Returns the length of this string.
boolean	<u>startsWith(String prefix)</u> Tests if this string starts with the specified prefix.
boolean	<u>startsWith(String prefix, int toffset)</u> Tests if the substring of this string beginning at the specified index starts with the specified prefix.
<u>String</u>	<u>substring(int beginIndex)</u> Returns a new string that is a substring of this string.
<u>String</u>	<u>substring(int beginIndex, int endIndex)</u> Returns a new string that is a substring of this string.

<u>String</u>	<u>toLowerCase()</u>
---------------	-----------------------------

Converts all of the characters in this `String` to lower case using the rules of the default locale.

<u>String</u>	<u>toUpperCase()</u>
---------------	-----------------------------

Converts all of the characters in this `String` to upper case using the rules of the default locale.

<u>String</u>	<u>trim()</u>
---------------	----------------------

Returns a copy of the string, with leading and trailing whitespace omitted.

static String valueOf (boolean b)	Returns the string representation of the <code>boolean</code> argument.
static String valueOf (char c)	Returns the string representation of the <code>char</code> argument.
static String valueOf (char[] data)	Returns the string representation of the <code>char</code> array argument.
static String valueOf (char[] data, int offset, int count)	Returns the string representation of a specific subarray of the <code>char</code> array argument.
static String valueOf (double d)	Returns the string representation of the <code>double</code> argument.
static String valueOf (float f)	Returns the string representation of the <code>float</code> argument.
static String valueOf (int i)	Returns the string representation of the <code>int</code> argument.
static String valueOf (long l)	Returns the string representation of the <code>long</code> argument.

Beispiel: Tiere

- In einem Zoo existieren Hasen, Füchse und Löwen
- Jedes dieser Tiere kann einen Laut von sich geben:
 - Hase: „br“
 - Fuchs: „brr“
 - Löwe: „brrrrr“
- Ein Tier frisst ein anderes Tier, wenn es lauter brüllen kann als das Andere

- Jedes Tier im Zoo hat einen Namen
- Es ist jederzeit bekannt, wie viele Tiere im Zoo leben
- Das Programm kann für jedes Tier entscheiden, ob es ein anderes Tier frisst oder nicht
- Außerdem kann es die Anzahl der Tiere im Zoo anzeigen