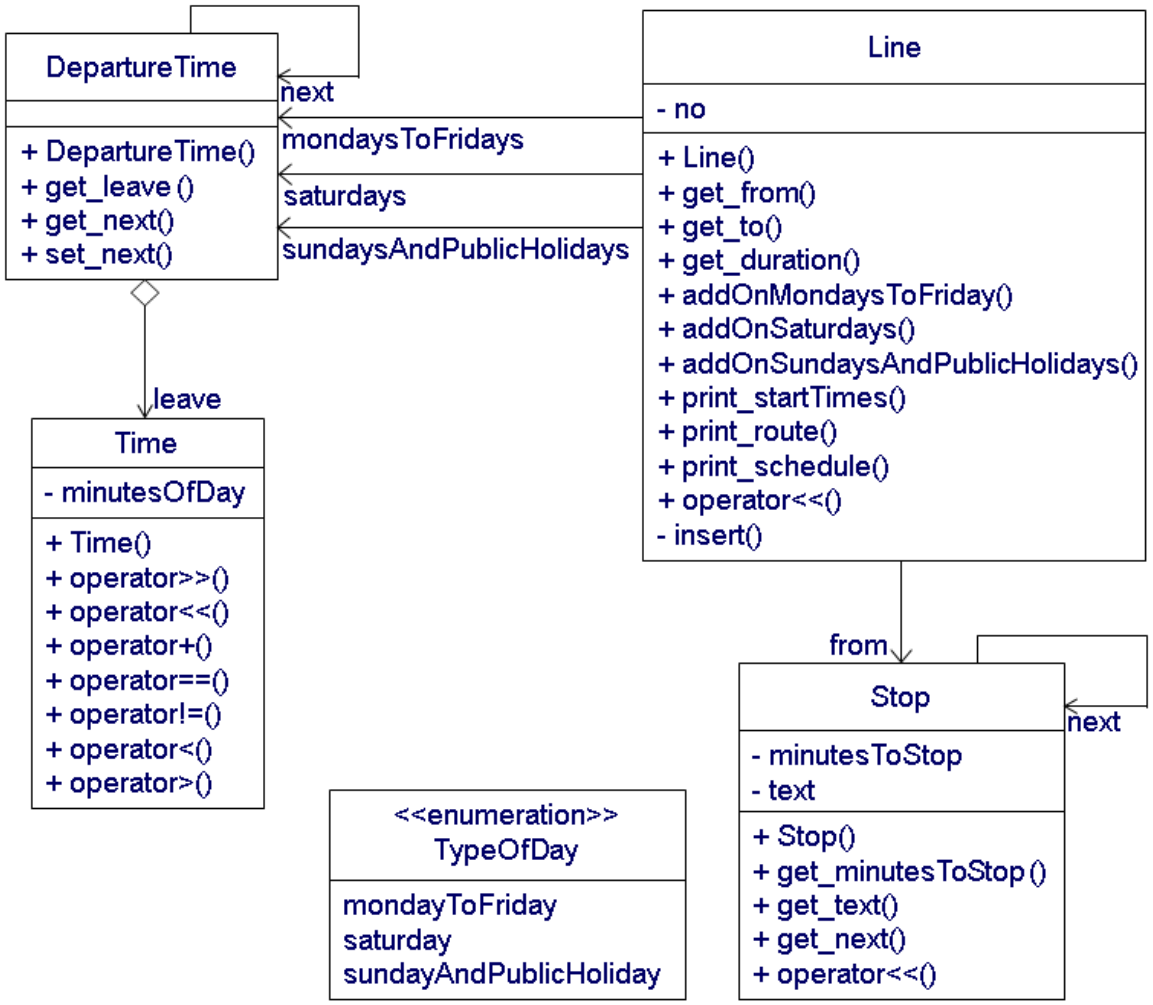
Public Transport Line Maps

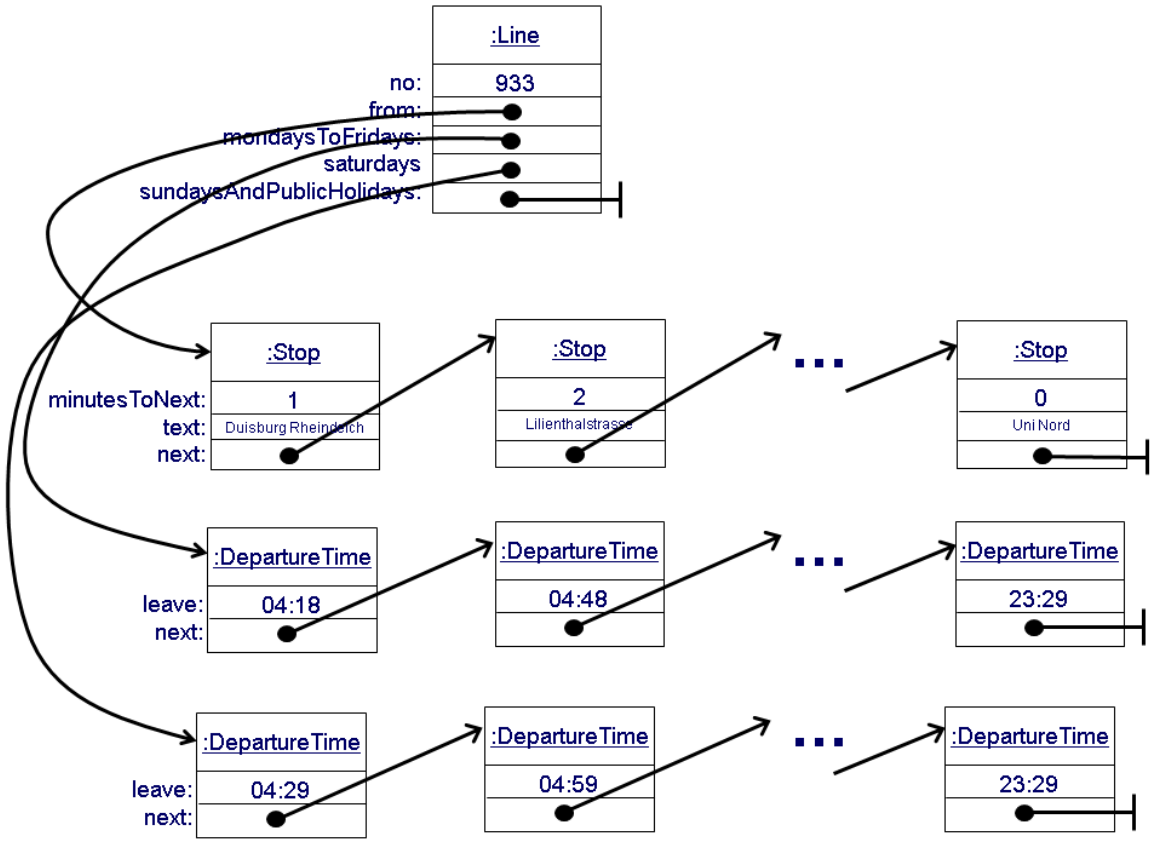
Overview

Like in many other domains digitalisation finds its way into the logistic area more and more. You have newly started a programmer job at Duisburg renowned company "Software Nerds". This company has a contract with the Duisburg Public Transport Company (DVG) to realise electronic (bus/tram/tube) line maps. In a first step you should write a C++ program with classes to efficiently store and print out line route and time tables in an object-oriented way.

The software architects have already developed below given UML class diagram (due to easy readability data types, parameters and destructors are left out) you should implement in C++ and test with example of bus line 933 (under https://www.dvg-duisburg.de/uploads/tx_moveelevatordvgbooklet/Internet_DVG_Lile_933_Gesamt.pdf you can find the actual data of this line).



Example Objects and Pointers for Bus Line 933:



Subtask 1

Define an enumeration class type with name **TypeOfDay** for the three to differentiate types of time schedules on mondays to fridays, saturdays as well as sundays and public holidays.
Define related to it an overloaded output operator **<<** printing **"monday to friday"**, **"saturday"** respectively **"sundays and public holidays**

Subtask 2

Define a class with name Time to store the clock time in a compact way. This class shall have following members:
• a private unsigned integer attribute with name minutesOfDay for the clock time stored in minutes of a day starting at 00:00 o'clock (e.g. 34 for 00:34, 594 for 09:54 oder 1389 for 23:09).

• a public standard constructor initialising 00:00 o'clock.
• a public constructor with two unsigned integer parameters for a hour and a minute able to initialise time hh:mm.
• a public friend binary input operator **>>** inputting a clock time by a to input value for the hour and a second to input value for the minute.

• a public friend binary output operator **<<** outputting a time always with five digits in form **hh:mm**.
• a public member addition operator **+** with one unsigned integer parameter for minutes returning a new clock time with added minutes (e.g. 07:13 + 5 = 07:18, 10:57 + 4 = 11:01, 23:59 + 3 = 00:02).

• a public binary member comparison operator **==** with one reference parameter to an object of this class comparing both clock times.
• a public binary member comparison operator **!=** with one reference parameter to an object of this class comparing both clock times
• a public binary member comparison operator **<** with one reference parameter to an object of this class comparing both clock times
• a public binary member comparison operator **>** with one reference parameter to an object of this class comparing both clock times.

All comparison operators return a Boolean value.

Subtask 3
Define a class with name Stop with data describing a (next) stop used to describe a list of stops. This class shall have following members:

• a private C++ string attribute with name **text** storing the description of a stop.
• a private attribute with name **next** of type pointer to **Stop** for building up a list of stops of a route
• a private attribute with name **next** of type pointer to **Stop** for building up a list of stops of a route
• a public constructor with three parameters for the number of minutes, the description and the next stop of a route with a null pointer as default parameter initialising the three private attributes.
• a public method with name get_minutesToNext without parameters returning the number of minutes driving time to this stop.
• a public method with name get_text without parameters returning the description of the stop.
• a public method with name get_next without parameters returning the next stop in a route.

Subtask 4
Define a class with name DepartureTime used to store a list of departure times of a (bus/tram/tube) line. This class shall have following members:

• a private attribute of type **Time** with name **leave** for a departure clock time.
• a private attribute with name **next** of type pointer to **DepartureTime** for building up a list of departure times of a route.
• a public constructor with two parameters for a departure time and a pointer to a next departure time having a null pointer as default parameter and intialising the two private attributes.

• a public method with name get_leave without parameters returning a reference to the attribute **leave** for the departure time.
• a public method with name get_next without parameters returning the next departure time of a line.
• a public method with name set_next with a pointer to an object of this class as parameter assigning it as next departure time of a line.

Subtask 5

Define a class with name Line for a (bus/tram/tube) line. This class shall have following members:
• a private unsigned integer attribute with name **no** storing the number of the line.
• a private pointer attribute with name **from** of class **Stop** for the head of a list of stops of this line.
• three private pointer attributes with names **mondaysToFridays**, **saturdays** and **sundaysAndPublicHolidays** of class **DepartureTime** storing the heads of three lists of departure times of this line.
• a private method with name insert with a (departure) time and a pointer of class **DepartureTime** as two parameters. The second parameter shall be head of a list of departure times into which the first given (departure) time shall be inserted. The new head of the list shall be returned as function value.
• a public constructor with two parameters for the number of the line and a pointer to the head of a list of stops to initialise the two related attributes. The three pointers for the departure times shall get initialised with null pointers.
• a public method with name get_from without parameters returning the description of the first departure stop.
• a public method with name get_to without parameters returning the description of the last arrival stop.
• a public method with name get_duration without parameters adding and returning the total driving time in minutes of the line route.
• three public methods with names **addOnMondaysToFridays**, **addOnSaturdays** and **addOnSundaysAndPublicHolidays** with each one departure time as object of class **Time** as parameter and without return values. Inside each function body as only statement appropriately call above defined private member function **insert** with fitting parameters to sorted insert the given parameter (departure) clock time into the respective list of departure clock times.
• a public method with name **print_startTimes** and an object of enumeration class **TypeOfDay** as parameter printing the sorted list of departure times for this type of day.
• a public method with name **print_route** without parameter printing the list of stops of the line. For each stop shall be outputted right adjusted in two characters the number of minutes to drive to this stop from the first stop and its description. In front of the first stop no minutes shall be printed out (see examples below).
• a public method with name **print_schedule** with the type of a day and a clock time object as both solely given parameters. In case of the line departs at this type of day and clock time a schedule with departure times at each stop shall be printed out (see examples below).
• a public friend binary output operator **<<** writing **"line: "** and the number of the line,

"from: " and the departing stop, **"to: "** and the arrival stop as well as **"duration: "** and the duration in minutes onto the given reference parameter output stream (by calling above member functions; see examples below).

Subtask 6

Define a global function returning a pointer to an object of class **Line**. In its body as example a new object on the heap for line 933 of DVG shall be defined as shown in the example above. The stops for this line are (here total drive time is outputted, to be stored are the driving time minutes to each next stop; shorter driving times in early mornings, late evenings, etc. shwon in time schedules shall be ignored):

For departure times as to be regard examples below two lists with data for Monday to Friday and on Saturday are sufficient:

montags bis freitags

Dulsburg Rheindelch (Bstg 1)	ab	4.18	4.48	5.18	5.48	6.03	6.18	▼	18.18	18.33	18.48	19.03	19.29	▼	21.29	21.59	22.29	22.59	23.29
- Lilienthalstraße (Bstg 1)		19	49	19	49	04	19		19	34	49	04	30		30	22.00	30	23.00	30
- Javastraße (Bstg 1)		21	51	21	51	06	21		21	36	51	06	31		31	01	31	01	31
- Katholische Kirche (Bstg 1)		22	52	22	52	07	22		22	37	52	07	32		32	02	32	02	32
- Am Schlütershof (Bstg 1)		23	53	23	53	08	23		23	38	53	08	32		32	02	32	02	32
- Tierhelm (Bstg 1)		24	54	24	54	09	24		24	39	54	09	33		33	03	33	03	33
- Sperrschleuse (Bstg 1)		26	56	26	56	11	26		26	41	56	11	35		35	05	35	05	35
- Landesarchiv NRW (Bstg 3)		28	58	28	58	13	28		28	43	58	13	37		37	07	37	07	37
- Schwanentor (Bstg 1)		30	5.00	30	6.00	15	30	alle	30	45	19.00	15	38	alle	38	08	38	08	38
- Friedrich-Wilhelm-Platz (Bstg 1)		33	03	33	03	18	33		33	48	03	18	40		40	10	40	10	40
- Lehmbruck-Museum (Bstg 1)		35	05	35	05	20	35	15	35	50	05	20	42	30	42	12	42	12	42
- Tonhallenstraße (Bstg 1)		36	06	36	06	21	36		36	51	06	21	43		43	13	43	13	43
- Dulsburg Hbf (Bstg 5)		38	08	38	08	23	38	Min.	38	53	08	23	45	Min.	45	15	45	15	45
- DU-Hbf Osteingang (Bstg 1)		40	10	40	10	25	40		40	55	10	25	47		47	17	47	17	47
- Blumenstraße (Bstg 1)		41	11	41	11	26	41		41	56	11	26	48		48	18	48	18	48
- Bismarckstraße (Bstg 3)		43	13	43	13	28	43		43	58	13	28	50		50	20	50	20	50
- Kammerstraße (Bstg 1)		45	15	45	15	30	45		45	19.00	15	30	51		51	21	51	21	51
- Lenaustraße (Bstg 1)		46	16	46	16	31	46		46	01	16	31	52		52	22	52	22	52
- Nettelbeckstraße (Bstg 2)		47	17	47	17	32	47		47	02	17	32	52		52	22	52	22	52
- Bürgerstraße (Bstg 2)		48	18	48	18	33	48		48	03	18	33	53		53	23	53	23	53
- Universität (Bstg 3)		49	19	49	19	34	49		49	04	19	34	54		54	24	54	24	54
- Uni Nord (Bstg 3)	an	4.50	5.20	5.50	6.20	6.35	6.50	▲	18.50	19.05	19.20	19.35	19.55	▲	21.55	22.25	22.55	23.25	23.55

samstags

Dulsburg Rheindelch (Bstg 1)	ab	4.29	4.59	5.29	5.59	6.29	6.48	▼	15.18	15.48	16.29	16.59	17.29	▼	21.29	21.59	22.29	22.59	23.29
- Lilienthalstraße (Bstg 1)		30	5.00	30	6.00	30	49		19	49	30	17.00	30		30	22.00	30	23.00	30
- Javastraße (Bstg 1)		31	01	31	01	31	51		21	51	31	01	31		31	01	31	01	31
- Katholische Kirche (Bstg 1)		32	02	32	02	32	52		22	52	32	02	32		32	02	32	02	32
- Am Schlütershof (Bstg 1)		32	02	32	02	32	53		23	53	32	02	32		32	02	32	02	32
- Tierhelm (Bstg 1)		33	03	33	03	33	54		24	54	33	03	33		33	03	33	03	33
- Sperrschleuse (Bstg 1)		35	05	35	05	35	56		26	56	35	05	35		35	05	35	05	35
- Landesarchiv NRW (Bstg 3)		37	07	37	07	37	58		28	58	37	07	37		37	07	37	07	37
- Schwanentor (Bstg 1)		38	08	38	08	38	7.00	alle	30	16.00	38	08	38	alle	38	08	38	08	38
- Friedrich-Wilhelm-Platz (Bstg 1)		40	10	40	10	40	03		33	03	40	10	40		40	10	40	10	40
- Lehmbruck-Museum (Bstg 1)		42	12	42	12	42	05	30	35	05	42	12	42	30	42	12	42	12	42
- Tonhallenstraße (Bstg 1)		43	13	43	13	43	06		36	06	43	13	43		43	13	43	13	43
- Dulsburg Hbf (Bstg 5)		45	15	45	15	45	08	Min.	38	08	45	15	45	Min.	45	15	45	15	45
- DU-Hbf Osteingang (Bstg 1)		47	17	47	17	47	10		40	10	47	17	47		47	17	47	17	47
- Blumenstraße (Bstg 1)		48	18	48	18	48	11		41	11	48	18	48		48	18	48	18	48
- Bismarckstraße (Bstg 3)		50	20	50	20	50	13		43	13	50	20	50		50	20	50	20	50
- Kammerstraße (Bstg 1)		51	21	51	21	51	15		45	15	51	21	51		51	21	51	21	51
- Lenaustraße (Bstg 1)		52	22	52	22	52	16		46	16	52	22	52		52	22	52	22	52
- Nettelbeckstraße (Bstg 2)		52	22	52	22	52	17		47	17	52	22	52		52	22	52	22	52
- Bürgerstraße (Bstg 2)		53	23	53	23	53	18		48	18	53	23	53		53	23	53	23	53
- Universität (Bstg 3)		54	24	54	24	54	19		49	19	54	24	54		54	24	54	24	54
- Uni Nord (Bstg 3)	an	4.55	5.25	5.55	6.25	6.55	7.20	▲	15.50	16.20	16.55	17.25	17.55	▲	21.55	22.25	22.55	23.25	23.55

Hint: to build up the lists for the departure times you may define appropriate small loops regarding the hour values and call in its bodies especially for the times every 15 or 30 minutes the member functions **addOnMondaysToFridays** or **addOnSaturdays**

Subtask 7

Write a **main** function with following definitions and statements:

• define for the whole public transport Rhein Ruhr area (VRR) an array with 1000 pointers to objects of class **Line** initialised with null pointers.
• call the function from sub task before and assign the returned pointer to array element with index 933.
• output the object, array element 933 points to, onto standard output stream using the output operator **<<**.
• send to the object, array element 933 points to, a message **print_route** to output its route.

• send to the object, array element 933 points to, a message **print_startTimes** to output its departure times mondays to fridays.
• send to the object, array element 933 points to, a message **print_startTimes** to output its departure times saturdays.
• input in a loop a line number, a type of day and a time, and send to the object, the array element with the inputted line number index points to, a message **print_schedule** to output the schedule for this time (see examples below).

Example Program Run

```
cout << line933 << endl;
line:     933
from:     Duisburg Rheindeich
to:       Uni-Nord
duration: 32 min

route of line 933
   Duisburg Rheindeich
 1 Lilienthalstrasse
 3 Javastrasse
 4 Katholische Kirche
 5 Am Schluetershof
 6 Tierheim
 8 Sperrschleuse
10 Landesarchiv NRW
12 Schwanentor
15 Friedrich-Wilhelm-Platz
17 Lehmbruck-Museum
18 Tonhallenstrasse
20 Duisburg Hbf
22 Duisburg Hbf Osteingang
23 Blumenstrasse
25 Bismarckstrasse
27 Kammerstrasse
28 Lenaustrasse
29 Nettelbeckstrasse
30 Buergerstrasse
31 Universitaet
32 Uni-Nord

start times of tours on mondays to fridays: 04:18 04:48 05:18 05:48 06:03 06:18
06:33 06:48 07:03 07:18 07:33 07:48 08:03 08:18 08:33 08:48 09:03 09:18 09:33 09
:48 10:03 10:18 10:33 10:48 11:03 11:18 11:33 11:48 12:03 12:18 12:33 12:48 13:0
3 13:18 13:33 13:48 14:03 14:18 14:33 14:48 15:03 15:18 15:33 15:48 16:03 16:18
16:33 16:48 17:03 17:18 17:33 17:48 18:03 18:18 18:33 18:48 19:03 19:29 19:59 20
:29 20:59 21:29 21:59 22:29 22:59 23:29

start times of tours on saturdays: 04:29 04:59 05:29 05:59 06:29 06:48 07:18 07:
48 08:18 08:48 09:18 09:48 10:18 10:48 11:18 11:48 12:18 12:48 13:18 13:48 14:18
 14:48 15:18 15:48 16:29 16:59 17:29 17:59 18:29 18:59 19:29 19:59 20:29 20:59 2
1:29 21:59 22:29 22:59 23:29
```

```
which line (> 1000 end): 926

which type of day?
1 mondays to fridays
2 saturdays
3 sundays and public holidays
1
which time (hh and mm): 8 17
sorry no line data

which line (> 1000 end): 933

which type of day?
1 mondays to fridays
2 saturdays
3 sundays and public holidays
1
which time (hh and mm): 9 18

line: 933, service on monday to friday
Duisburg Rheindeich              09:18
Lilienthalstrasse                09:19
Javastrasse                      09:21
Katholische Kirche               09:22
Am Schluetershof                 09:23
Tierheim                         09:24
Sperrschleuse                    09:26
Landesarchiv NRW                 09:28
Schwanentor                      09:30
Friedrich-Wilhelm-Platz          09:33
Lehmbruck-Museum                 09:35
Tonhallenstrasse                 09:36
Duisburg Hbf                     09:38
Duisburg Hbf Osteingang          09:40
Blumenstrasse                    09:41
Bismarckstrasse                  09:43
Kammerstrasse                    09:45
Lenaustrasse                     09:46
Nettelbeckstrasse                09:47
Buergerstrasse                   09:48
Universitaet                     09:49
Uni-Nord                         09:50
```

```
which line (> 1000 end): 933

which type of day?
1 mondays to fridays
2 saturdays
3 sundays and public holidays
2
which time (hh and mm): 16 48

line: 933, service on saturday
no service at 16:48

which line (> 1000 end): 933

which type of day?
1 mondays to fridays
2 saturdays
3 sundays and public holidays
2
which time (hh and mm): 13 48

line: 933, service on saturday
Duisburg Rheindeich             13:48
Lilienthalstrasse               13:49
Javastrasse                     13:51
Katholische Kirche              13:52
Am Schluetershof                13:53
Tierheim                        13:54
Sperrschleuse                   13:56
Landesarchiv NRW                13:58
Schwanentor                     14:00
Friedrich-Wilhelm-Platz         14:03
Lehmbruck-Museum                14:05
Tonhallenstrasse                14:06
Duisburg Hbf                    14:08
Duisburg Hbf Osteingang         14:10
Blumenstrasse                   14:11
Bismarckstrasse                 14:13
Kammerstrasse                   14:15
Lenaustrasse                    14:16
Nettelbeckstrasse               14:17
Buergerstrasse                  14:18
Universitaet                    14:19
Uni-Nord                        14:20
```