

1 Problem Overview

- Sort and order packages onto truck
- Determine packages that can be delivered on time.
 - ◊ Driver has 8 hours to return to warehouse.
- Determine and deliver driving directions from currently location to driver for next address.

2 Classes

2.1 Package

- UUID
 - ◊ Each package exists as **one package object** in the system
- Package pointers to two client objects
 - ◊ Sender Client
 - ◊ Receiver Client
- Structure to hold packages currently
 - ◊ In the warehouse
 - ◊ On the truck
- Packages have a priority
 - ◊ Overnight
 - ◊ Two day
 - ◊ Regular

2.2 Clients

- Separate client information from packages
- Client object should represent one physical client.
- Each client (sender/receiver) points to one or multiple packages.
 - ◊ Separate sent package pointers
 - ◊ Separate received package pointers

2.3 Trucks

- Trucks have a weight limit.
- Trucks, and drivers, have 8 hours to return to the warehouse after start.

2.4 City

- City is divided into quadrants.
 - ◊ Streets North of center (Main Street) are numbered sequentially (1st Street North, 2nd Street North, etc.)
 - ◊ Streets South of center (Main Street) are numbered as (1st Street South, 2nd Street South, etc.)
 - ◊ Avenues West of center (Central Avenue) are numbered as (1st Avenue West, etc)
 - ◊ Similar for East of Central Avenue.
- Assume truck can drive one block per minute.
 - ◊ Stops take 5 minutes.

3 Program Overview

- Make file

3.1 Structures

3.1.1 Map

- Collection of nodes with edge weights as grid driving distance (truck and only go north/south or east/west.)

3.1.2 Packages

- Age variable for two-day or regular packages.
 - ◊ Birthdate or mail dates.
 - ◊ Prevents late packages.

3.2 Style

- Classes (method) and object names are initial capitalized (MyClass)
- Class member variables are initial lowercase (myVar)
- All variables, functions, methods, etc are camelCase.
- Following Qt Doxygen style (see below.) Doxygen documentation <http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html>

```
1  //! A test class.
2  /*!
3   A more elaborate class description.
4  */
5  class QTeststyle_Test
6  {
7      public:
8          //! An enum.
9          /*! More detailed enum description. */
10         enum TEnum {
11             TVal1, /*!< Enum value TVal1. */
12             TVal2, /*!< Enum value TVal2. */
13             TVal3 /*!< Enum value TVal3. */
14         }
15         /*! Enum pointer.
16         /*! Details. */
17         *enumPtr,
18         /*! Enum variable.
19         /*! Details. */
20         enumVar;
21
22         /*! A constructor.
23         /*!
24         A more elaborate description of the constructor.
25         */
26         QTeststyle_Test();
27         /*! A destructor.
28         /*!
29         A more elaborate description of the destructor.
30         */
31         ~QTeststyle_Test();
32
33         /*! A normal member taking two arguments and returning an
34         integer value.
35         /*!
36         \param a an integer argument.
37         \param s a constant character pointer.
38         \return The test results
39         \sa QTeststyle_Test(), ~QTeststyle_Test(), testMeToo() and
40         publicVar()
41         */
42         int testMe(int a, const char *s);
43
44         /*! A pure virtual member.
```

```
43  /*!  
44  \sa testMe()  
45  \param c1 the first argument.  
46  \param c2 the second argument.  
47  */  
48  virtual void testMeToo(char c1, char c2) = 0;  
49  
50  //! A public variable.  
51  /*!  
52  Details.  
53  */  
54  int publicVar;  
55  
56  //! A function variable.  
57  /*!  
58  Details.  
59  */  
60  int (*handler)(int a, int b);  
61  };
```

3.3 Classes

- Client class
- Package class
- Truck class
 - ◊ Weight initialize
 - ◊ Trucks are contained in a vector
 - Multiple trucks are passed to other class objects
- Map class
 - ◊ Creates adjacency list of packages.
- Routing class (algorithm)
 - ◊ Truck vector is passed to the algorithm
 - ◊ Map graph adjacency list is passed to the algorithm
 - ◊ Routing algorithm loads truck based on “something”

3.4 Input Files

Comma separated list of packages to deliver.

- Each line includes the sender and receiver.