

## Отчет по 2-й лабораторной работе

ЛР выполнили: Прокопенко Дарья, Герасимова Елизавета, Аргичев Дмитрий;  
гр.М30-406с-19.

Цель работы:

1) Разработать программу, в которой будет реализован следующий функционал:

а) Класс комплексной навигационной системы, которая включает в себя:

- name – наименование навигационной системы;
- коллекция датчиков;
- add\_sensor – метод добавляющий датчик к навигационной системе;
- measure\_acc – метод возвращающий измерения от всех акселерометров;
- measure\_gyro – метод возвращающий измерения от всех гироскопов;
- measure\_position – метод возвращающий измерения от всех датчиков положения;
- list\_sensors – выводит имена датчиков входящих в состав навигационной системы;
- list\_sensors\_info – выводит имена датчиков входящих в состав навигационной системы;

б) Каждый датчик - это отдельный объект. Датчик обладает следующими свойствами и методами:

- name – содержит имя датчика;
- measurement\_unit – единицей измерения того что он измеряет;
- measurement\_range – пара минимального и максимального измеряемого значения;
- measure – метод генерирующий случайное значение с плавающей точкой в диапазоне значений измерителя.

2) Собрать программу через консоль PowerShell, используя компилятор установленный через MinGW. При компиляции обязательно использовать флаги -Wall и -Werror.

3) Написать Makefile для сборки проекта.

Ход работы:

1) Разработка программы.

а) Изначально был создан класс комплексной навигационной системы, которая включала в себя следующие поля и методы:

1. name – наименование навигационной системы

2. коллекция датчиков.

2. `add_sensor` – метод добавляющий датчик к навигационной системе.

3. `measure_acc` – метод возвращающий измерения от всех акселерометров.

4. `measure_gyro` – метод возвращающий измерения от всех гироскопов.

5. `measure_position` – метод возвращающий измерения от всех датчиков положения.

6. `list_sensors` – выводит имена датчиков входящих в состав навигационной системы.

б) Также был добавлен метод `sensor_measurement_list`, который выводил число в диапазоне, соответствующем определенному датчику. Сами датчики были описаны как массив объекта класса `sensor`.

В класс `sensor` входят следующие поля и методы:

1. `name` – содержит имя датчика.

2. `measurement_unit` – единицей измерения того, что он измеряет.

3. `measurement_range` – пара минимального и максимального измеряемого значения.

4. `measure` – метод генерирующий случайное значение с плавающей точкой в диапазоне значений измерителя.

Поле `measurement_range` описали как объект класса `range`, в котором хранились две переменные: `min` и `max`. В методе `measure` при описании диапазона, в котором будет генерироваться число, обращались к паре минимального и максимального значения.

В `main` создали объект класса `complex_navigation_system` и вызвали функции `add_sensor`, `list_sensors`, `sensor_measurement_list`.

2) После разработки программы она была собрана через консоль PowerShell, используя компилятор MinGW с добавлением флагов `-Wall` и `-Werror`. Для этого требовалось:

- В консоли PowerShell указать расположение файлов, для этого использовалась команда `cd D:\TechProLR2;`

- После чего скомпилировать файл `cpp` командой `g++ -Wall -Werror techprol.cpp -o name`, где `name`-это название `exe` файла;

- Запустить `exe` файл командой `.\name.exe`.

3) После сборки программы был создан `Makefile`.

Makefile — это файл с инструкциями для утилиты make, которая нужна для автоматической сборки проекта.

Пример содержания Makefile:

```
all:

    g++ -Wall -Werror techprol.cpp -o techprol
```

где, all-это цель, после которой указаны нужные команды, с использованием табуляции.

Для того, чтобы запустить Makefile была использована команда `make -f nameMakefile`, где nameMakefile-имя файла.

Работа в бригаде была распределена следующим образом:

- Герасимова Елизавета: создание "шаблона" программы, описание методов `add_sensor`, `list_sensors`, `measure_acc`.
- Прокопенко Дарья: описание классов `complex_navigation_system`, `sensor`, описание метода `measure_gyro`.
- Аргичев Дмитрий: описание методов `sensor_measurement_list`, `measure_position`.

Вывод:

В ходе данной лабораторной работы было выполнено:

- 1) Разработка программы в VS Code, используя созданный репозиторий в GitHub для совместной работы;
- 2) Сборка программы через консоль PowerShell, используя компилятор установленный через MinGW. При компиляции были использованы флаги `-Wall` и `-Werror`;
- 3) Написан Makefile для сборки проекта.