|  |  |
| --- | --- |
| ДИСЦИПЛИНА | Технологии индустриального программирования |
| ИНСТИТУТ | ИПТИП |
| КАФЕДРА | Индустриального программирования |
| ВИД УЧЕБНОГО МАТЕРИАЛА | Методические указания к практическим занятиям по дисциплине |
| ПРЕПОДАВАТЕЛЬ | Александров Алексей Сергеевич |
| СЕМЕСТР | 3 семестр, 2024/2025 уч. год |

# Практическое занятие №14

## Множества и хеш-таблицы в Qt.

Работа с ассоциативными контейнерами будет рассмотрена на примере контейнера Map, представленного в Qt классом QMap.

Map представляет из себя таблицу вида «ключ – значение». Ключом и значением могут быть любые типы данных. Ключ должен быть уникальный, а значения могут повторяться, но одному ключу соответствует только одно значение, т.е. при добавлении нового значения с уже существующим ключом, старое значение будет перезаписано.

При создании объекта QMap, необходимо указать в <> тип данных ключа и значения:

QMap<QString, int> agesMap; // ключ – строка, значение – число

Добавление данных производится с помощью оператора []:

map["Иванов"] = 20;

map["Петров"] = 24;

map["Данилов"] = 35;

Аналогично значения можно добавлять с помощью метода insert():

map.insert("Иванов", 20);

map.insert("Петров", 24);

map.insert("Данилов", 35);

Получить значение можно также с помощью оператора [] или методом value():

int age1 = map["Иванов"];

int age2 = map.value("Петров");

В случае отсутствия заданного ключа, будет возвращено значение по умолчанию для данного типа данных.

Проверить, существует ли ключ в контейнере можно с помощью метода contains():

int age = 0;

if (map.contains("Иванов"))

age = map.value("Иванов");

Также, можно указать, какое значение требуется вернуть, если ключ отсутствует:

int age = map.value("Иванов", 30);

В данном случае, если не будет найден ключ «Иванов», в переменную age будет записано число 30.

За счёт этого, в коде лучше использовать метод value(), т.к. его использование более безопасное, чем оператор [].

Перебор значений в QMap возможен с помощью итератора в Java-Style виде:

QMapIterator<QString, int> i(map);

while (i.hasNext()) {

i.next();

cout << i.key() << ": " << i.value() << endl;

}

А также перебор в стиле стандартной С++ библиотеки STL:

for (auto i = map.cbegin(), end = map.cend(); i != end; ++i)

cout << i.key() << ": " << i.value() << endl;

Оба этих варианта эквиваленты, поэтому можно использовать любой из них.

Если же требуется перебрать только значение, без ключей, то можно использовать цикл for:

for (int value : std::as\_const(map))

cout << value << endl;

Удалить элемент из контейнера можно с помощью метода remove():

map.remove("Иванов");

Таким образом были рассмотрены основные методы работы с контейнером QMap.