УНИВЕРСИТЕТ СИНЕРГИЯ

ВЕБИНАР 4

КОНТЕЙНЕРЫ QSET, QMAP.

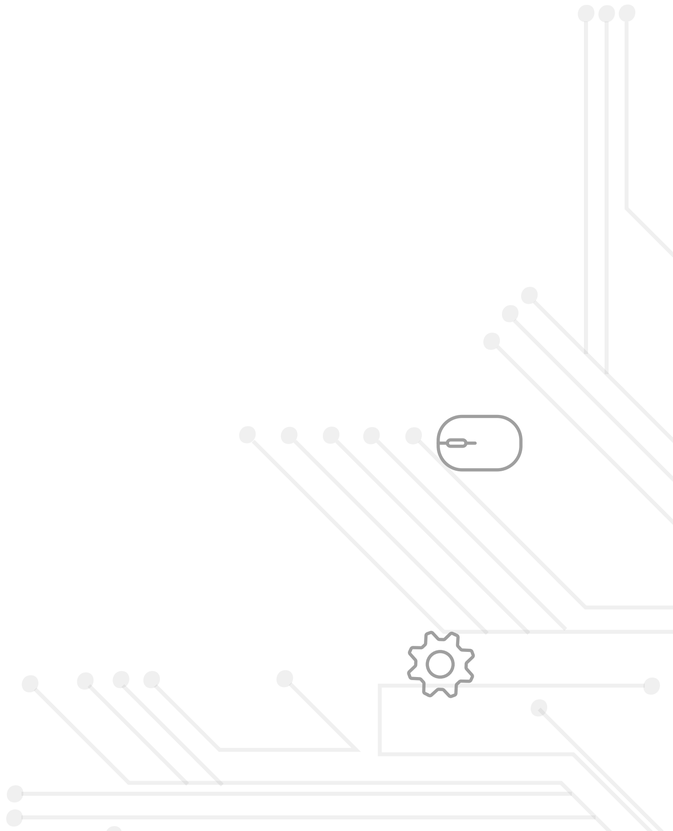
Теория

Контейнеры QSet и QMap являются частями библиотеки Qt, которая предоставляет инструменты для разработки приложений на C++.

Оба контейнера (QSet и QMap) предоставляют широкий набор методов для работы с элементами, включая добавление, удаление, обновление и поиск. Они являются полезными инструментами при разработке приложений, где требуется хранение коллекции уникальных значений или пар ключ-значение.

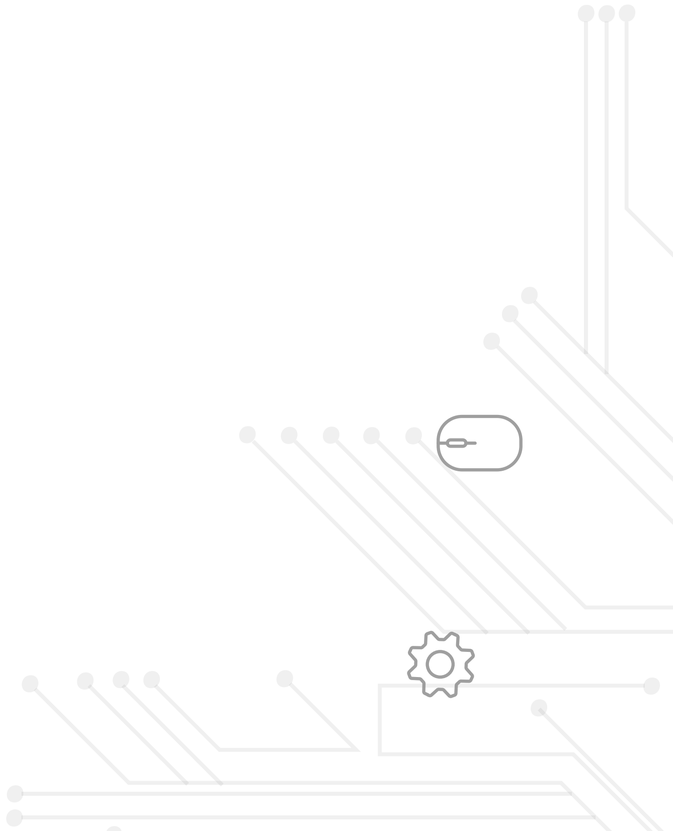
Контейнер QSet

QSet - это контейнер, который представляет собой множество уникальных значений. Он хранит элементы в отсортированном порядке и обеспечивает быстрый доступ к элементам с помощью хэш-таблицы.



Функционал контейнера QSet

- QSet обеспечивает поиск, вставку, удаление и проверку наличия элементов за константное время O(1) в среднем случае. Кроме того, QSet имеет встроенную функцию сортировки элементов в порядке возрастания.
- QSet также предоставляет различные операции для объединения, пересечения и разности двух множеств, а также возможность преобразовать множество в QList или QVector.



Пример 1

```
#include <QSet>
#include <QDebug>
int main() {
    QSet<QString> set;
    set.insert("apple");
    set.insert("banana");
    set.insert("orange");
    qDebug() << "Set size:" << set.size();
// Выводит: "Set size: 3"
et.remove("banana");
    qDebug() << "Is 'apple' in set?" << set.contains("apple");
 // Выводит: "Is 'apple' in set? true"
    qDebug() << "Is 'banana' in set?" << set.contains("banana");
// Выводит: "Is 'banana' in set? false"
    for (const QString& item : set) {
        qDebug() << item;
  // Выводит: "apple", "orange"
    }
    return 0;
}
```

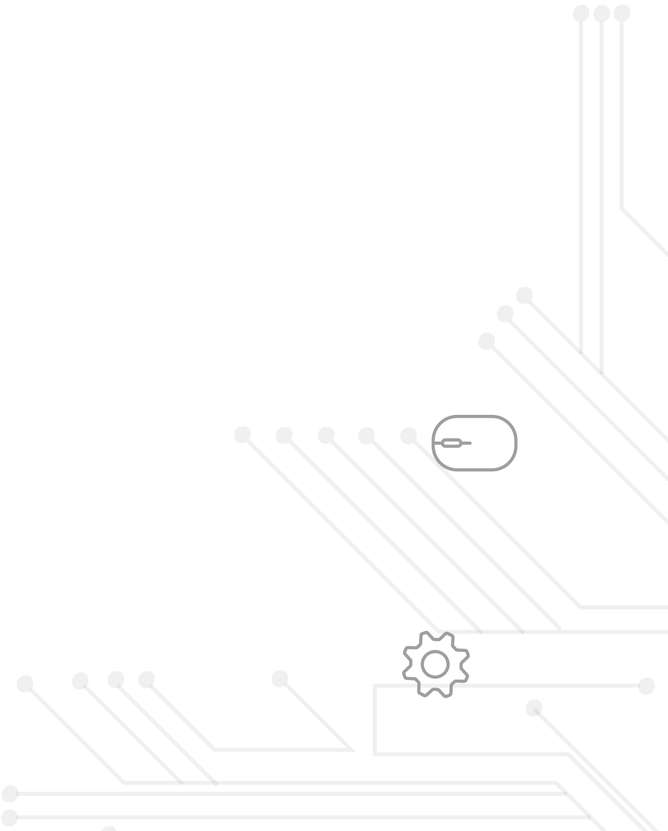
Пример 2

```
#include <QSet>
#include <QDebug>
int main() {
    QSet<int> numbers;
    numbers.insert(1);
    numbers.insert(2);
    numbers.insert(3);
qDebug() << "Size: " << numbers.size();
// Выводит: "Size:  3"
    qDebug() << "Contains 2: " << numbers.contains(2); // Выводит: "Contains 2:  true"
    qDebug() << "Contains 4: " << numbers.contains(4); // Выводит: "Contains 4:  false"
    return 0;
}
```

Пример 3

```
QSet<int> numbers;
numbers << 1 << 2 << 3 << 4;
if (numbers.contains(2)) {
    qDebug() << "2 is in the set";
}
```

```
qDebug() << "Set size:" << numbers.size();
qDebug() << "Set elements:";
for (int number : numbers) {
    qDebug() << number;
}
```



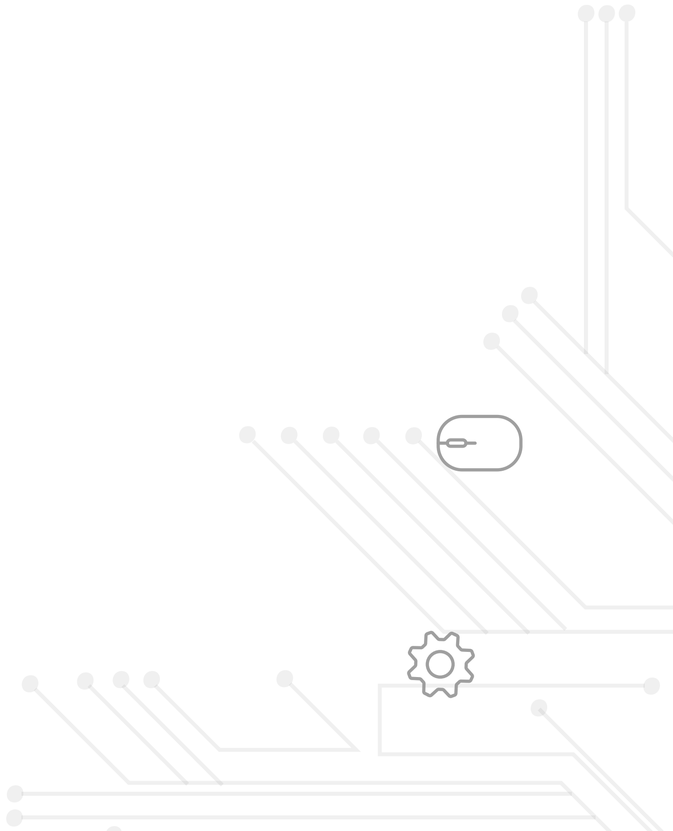
Контейнер QMap

QMap - это контейнер в языке программирования C++, который представляет собой отсортированный по ключу ассоциативный контейнер, где каждому уникальному ключу соответствует значение.

Контейнер QMap является реализацией ассоциативного массива на базе бинарного дерева поиска. Он обеспечивает высокую эффективность операций доступа к элементу и быстрое выполнение операций вставки, удаления и поиска.

Функционал контейнера QMap

Как и QSet, QMap предоставляет функции для работы с отображениями, такие как вставка, удаление и поиск элементов. Он также поддерживает итерацию по элементам и сортировку отображения по значению или пользовательским критериям.



Пример 1

```
#include <QMap>
#include <QString>
#include <QDebug>
int main() {
    QMap<QString, int> ages;
    ages.insert("Alice", 25);
    ages.insert("Bob", 30);
    ages.insert("Charlie", 35);

qDebug() << "Alice's age:" << ages.value("Alice"); // Выводит 25
    QMap<QString, int>::iterator i;
    for (i = ages.begin(); i != ages.end(); ++i) {
        qDebug() << i.key() << ":" << i.value();
    }
    return 0;
}
```

Выбор контейнера для проекта

Выбор контейнера зависит от конкретных требований вашего проекта и типа данных, с которыми вы работаете.

QSet используется, когда необходимо работать с множеством элементов, учитывая возможность записать элементы в некотором порядке и очень быстро просмотреть и найти значения, а также провести операции над множествами.

QMap используется, когда необходимо хранить элементы одного и того же типа, индексируемые ключевыми значениями.

Выбор контейнера для проекта

Задача 1

У вас есть список учеников класса, и вам необходимо поделить их на две категории: мальчики и девочки. Какой контейнер для решения данной задачи вы будете использовать?

Задача 2

Вы составили список продуктов для похода в магазин:

01 Молоко

02 Хлеб

03 Сыр

Какой контейнер вы будете использовать для быстрого поиска позиции из списка?