

Название класса	Пара (pair)	Массив (array)	Вектор (vector)	Список (list)
Какую библиотеку подключить	<utility>	<array>	<vector>	<list>
Синтаксис инициализации	std::pair myName("Ivan", "Ivanov");	std::array<int, 4> myarray = { 8, 6, 4, 1 };	std::vector<int> myvect = { 10, 8, 6, 4, 2, 1 };	std::list<int> numbers = { 1, 2, 3, 4, 5 };
Какими способами можно получить доступ к произвольному элементу класса?	В паре может быть всего два элемента. Доступ получается с помощью .first для первого элемента и .second для второго. Например: myName.first = "Ivan", myName.second = "Ivanov".	1) Доступ к значениям массива с помощью оператора индекса (как обычно): std::cout << "myarray[" << i << "]" = " << myarray[i] << "\n"; 2) Доступ к значениям массива с помощью функции at(), которая осуществляет проверку диапазона: myarray.at(1) = 7; // элемент массива 1 - корректный, присваиваем элементу массива 1 значение 7	Доступ к элементам массива может выполняться как через оператор [] (который не выполняет проверку диапазона), так и через функцию at() (которая выполняет проверку диапазона): myvect[7] = 3; // без проверки диапазона myvect.at(8) = 4; // с проверкой диапазона	Доступ к произвольному элементу - запрещен
Можно ли добавлять новые элементы? Если да, то каким образом? С какой стороны разрешено их добавлять?	Нельзя.	Добавлять новые элементы можно но, не должно превышать его размер. Пример: // объявление массива std::array myarray; // с помощью списка инициализаторов myarray = { 0, 1, 2, 3 }; //добавление нового элемента myarray.at(1) = 7;	Можно. Добавлять в конец вектора.	Можно. Пример: std::list<int> numbers = { 1, 2, 3, 4, 5 }; numbers.push_back(23); // { 1, 2, 3, 4, 5, 23 } добавляет значение в конец списка, начало, на позицию, на которую указывает итератор <b>pos</b>
Можно ли удалять элементы? Если да, то каким образом? С	Нельзя.	Нельзя.	Можно myvect.erase	Можно

какой стороны разрешено их удалять?				
Каким способом можно узнать количество элементов в классе?	Класс «пара» всегда подразумевает наличие двух элементов.	array.size(): возвращает размер контейнера.	Vector – знает свою длину. array.size():возвращает размер контейнера	Для получения размера списка можно использовать функцию size(). Функция empty() позволяет узнать, пуст ли список. Если он пуст, то функция возвращает значение true, иначе возвращается значение false.
Предлагает ли класс автоматическую сортировку данных? Если да, то какую?	Не предлагает.	Не предлагает.	Не предлагает.	Не предлагает.
При каком условии объект этого класса может содержать два или более одинаковых значений?	Всегда может.	Всегда может.	Всегда может.	Всегда может.
Какие специфичные функции можно применять к классу? Что они делают? (только из того, что мы изучали на лекции)	Функция std::make_pair(value, value) позволяет быстро создавать пары из двух значений.	Массив std::array<int, число> - автоматически после своего выполнения освобождает память	Изменение размера вектора - array.resize(число); Емкость вектора - array.capacity()	push_back(val): добавляет значение val в конец списка; push_front(val): добавляет значение val в начало списка;

Название класса	Односвязный список (forward list)	Дек (deque)	Словарь (map/multimap)	Набор (set/multiset)
Какую библиотеку подключить	<forward_list>	<deque>	<map>/< <b>multimap</b> >	<set>
Синтаксис инициализации	std::forward_list<int> numbers = { 1, 2, 3, 4, 5 };	std::deque<int> deque = { 1, 2, 3, 5 }; // deque состоит из чисел 1, 2, 3, 5	std::map<std::string, int> books; //Содержит связку название книги / количество	std::multiset<int> ms;
Какими способами можно получить доступ к произвольному элементу класса?	Напрямую в списке forward_list можно получить только первый элемент. Для этого применяется функция front().	[index]: получение элемента по индексу; at(index): возвращает элемент по индексу; front(): возвращает первый элемент;	С помощью итераторов	S.lower_bound(a) - первый элемент, меньший, чем a; S.upper_bound(a) - первый элемент, больший, чем a;

		back(): возвращает последний элемент.		
Можно ли добавлять новые элементы? Если да, то каким образом? С какой стороны разрешено их добавлять?	Для добавления элементов в forward_list применяются следующие функции: push_front(val): добавляет объект val в начало списка emplace_front(val): добавляет объект val в начало списка; emplace_after(iter, val): вставляет объект val после элемента, на который указывает итератор iter. Возвращает итератор на вставленный элемент. Если iter представляет итератор на позицию после конца списка, то результат неопределен; insert_after(iter, val): вставляет объект val после элемента, на который указывает итератор iter. Возвращает итератор на вставленный элемент. Если iter представляет итератор на позицию после конца списка, то результат неопределен;	Можно push_back(val): добавляет значение val в конец очереди; push_front(val): добавляет значение val в начало очереди;	Можно	Можно S.insert(a) - вставка элемента a;
Можно ли удалять элементы? Если да, то каким образом? С какой стороны разрешено их удалять?	Можно pop_front(): удаляет первый элемент; erase_after(iter): удаляет	Можно pop_back(): удаляет последний элемент pop_front(): удаляет первый элемент	Можно  erase(key) - удалить указанный элемент; erase(it), erase(start,end)	Можно S.erase(S2) - удаляет последовательность элементов;

	<p>элемент после элемента, на который указывает итератор <code>iter</code>.</p> <p>Возвращает итератор на элемент после удаленного;</p> <p><code>erase_after(begin, end)</code>: удаляет диапазон элементов, на начало и конец которого указывают соответственно итераторы <code>begin</code> и <code>end</code>.</p> <p>Удаление в середине односвязного списка может его разрушить</p>		- удаляет элемент с заданным итератором или между	
Каким способом можно узнать количество элементов в классе?	Функция <code>empty()</code> позволяет узнать, пуст ли список.	<code>std::deque&lt;int&gt; numbers = { 1, 2, 3, 4, 5 };</code> <code>numbers.size()</code>	<code>size()</code> - возвращает число элементов;	<code>S.size()</code> - размер контейнера;
Предлагает ли класс автоматическую сортировку данных? Если да, то какую?	Не предлагает.	Не предлагает.	Предлагает.	Предлагает.
При каком условии объект этого класса может содержать два или более одинаковых значений?	Всегда может.	Всегда может.	Всегда может. Содержит пара	Содержит пара
Какие специфичные функции можно применять к классу? Что они делают? (только из того, что мы изучали на лекции)	Функция <code>empty()</code> позволяет узнать, пуст ли список.	При удалении элемента в <b>deque</b> он полностью перезаписывается	У <b>map</b> есть один небольшой недостаток: в нём одному ключу нельзя поставить в соответствие несколько объектов. Решение <b>map - multimap</b>	