**Темы курсового проекта по ООП**

**Порядок выполнения работы.**

Написать и отладить три программы. Первая программа демонстрирует использование контейнерных классов для хранения встроенных типов данных.

Вторая программа демонстрирует использование контейнерных классов для хранения пользовательских типов данных.

Третья программа демонстрирует использование алгоритмов STL.

**В программе № 1** выполнить следующее:

1. Создать объект-контейнер в соответствии с вариантом задания и заполнить его данными, тип которых определяется вариантом задания.

2. Просмотреть контейнер.

3. Изменить контейнер, удалив из него одни элементы и заменив другие.

4. Просмотреть контейнер, используя для доступа к его элементам итераторы.

5. Создать второй контейнер этого же класса и заполнить его данными того же типа, что и первый контейнер.

6. Изменить первый контейнер, удалив из него n элементов после заданного и добавив затем в него все элементы из второго контейнера.

7. Просмотреть первый и второй контейнеры.

**В программе № 2** выполнить то же самое, но для данных пользовательского типа.

**В программе № 3** выполнить следующее:

1. Создать контейнер, содержащий объекты пользовательского типа. Тип контейнера выбирается в соответствии с вариантом задания.

2. Отсортировать его по убыванию элементов.

3. Просмотреть контейнер.

4. Используя подходящий алгоритм, найти в контейнере элемент, удовлетворяющий заданному условию.

5. Переместить элементы, удовлетворяющие заданному условию в другой (предварительно пустой) контейнер. Тип второго контейнера определяется вариантом задания.

6. Просмотреть второй контейнер.

7. Отсортировать первый и второй контейнеры по возрастанию элементов.

8. Просмотреть их.

9. Получить третий контейнер путем слияния первых двух.

10. Просмотреть третий контейнер.

11 .Подсчитать, сколько элементов, удовлетворяющих заданному условию, содержит третий контейнер.

12.Определить, есть ли в третьем контейнере элемент, удовлетворяющий заданному условию.

**Методические указания**

1. В качестве пользовательского типа данных использовать пользовательский класс (см. Приложение 1).

2. При создании контейнеров в программе № 2 объекты загружать из потока.

3. Для вставки и удаления элементов контейнера в программе № 2 использовать соответствующие операции, определенные в классе контейнера.

4. Для создания второго контейнера в программе № 3 можно использовать либо алгоритм **remove\_copy\_if**, либо определить свой алгоритм **copy\_if**, которого нет в STL.

5. Для поиска элемента в коллекции можно использовать алгоритм **find\_if**, либо **for\_each**, либо **binary\_search**, если контейнер отсортирован.

6. Для сравнения элементов при сортировке по возрастанию используется операция <, которая должна быть перегружена в пользовательском классе. Для сортировки по убыванию следует написать функцию **comp** и использовать вторую версию алгоритма **sort.**

7. Условия поиска и замены элементов выбираются самостоятельно и для них пишется функция-предикат.

8. Для ввода-вывода объектов пользовательского класса следует перегрузить операции “>>” и “<<”.

9. Некоторые алгоритмы могут не поддерживать используемые в вашей программе контейнеры. Например, алгоритм **sort** не поддерживает контейнеры, которые не имеют итераторов произвольного доступа. В этом случае следует написать свой алгоритм. Например, для стека алгоритм сортировки может выполняться следующим образом: переписать стек в вектор, отсортировать вектор, переписать вектор в стек.

10.При перемещении элементов ассоциативного контейнера в неассоциативный перемещаются только данные (ключи не перемещаются). И, наоборот, при перемещении элементов неассоциативного контейнера в ассоциативный должен быть сформирован ключ.

**Варианты заданий**

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Первый  контейнер | Второй  контейнер | Встроенный  тип данных |
| 1 | vector | list | int |
| 2 | list | deque | long |
| 3 | deque | stack | float |
| 4 | stack | queue | double |
| 5 | queue | vector | char |
| 6 | vector | stack | string |
| 7 | map | list | long |
| 8 | multimap | deque | float |
| 9 | set | stack | int |
| 10 | multiset | queue | char |
| 11 | vector | map | double |
| 12 | list | set | int |
| 13 | deque | multiset | long |
| 14 | stack | vector | float |
| 15 | queue | map | int |
| 16 | priority\_queue | stack | char |
| 17 | map | queue | char |
| 18 | multimap | list | int |
| 19 | set | map | char |
| 20 | multiset | vector | int |
| 21 | vector | map | double |
| 22 | vector | list | double |
| 23 | map | queue | long |
| 24 | set | map | long |
| 25 | set | stack | double |
| 26 | stack | queue | string |
| 27 | list | set | string |
| 28 | vector | map | string |
| 29 | list | deque | string |
| 30 | queue | map | double |
| 31 | multimap | deque | double |

**Приложение 1**

1. СТУДЕНТ

имя – string

курс – int

пол – int(bool)

2. ИЗДЕЛИЕ

имя – string

шифр – string

количество – int

3. АДРЕС

имя – string

улица – string

номер дома – int

4. ЦЕХ

имя – string

начальник – string

количество

работающих – int

5. СТРАНА

имя – string

форма

правления – string

площадь – float

6. СЛУЖАЩИЙ

имя – string

возраст – int

рабочий стаж – int

7. БИБЛИОТЕКА

имя – string

автор – string

стоимость – float

8. ТОВАР

имя – string

количество – int

стоимость – float

9. ПЕРСОНА

имя – string

возраст – int

пол – int(bool)

10. ЖИВОТНОЕ

имя – string

класс – string

средний вес – int

11. КАДРЫ

имя – string

номер цеха – int

разряд – int

12. ЭКЗАМЕН

имя студента – string

дата – int

оценка – int

13. КВИТАНЦИЯ

номер – int

дата – int

сумма – float

14. АВТОМОБИЛЬ

марка – string

мощность – int

стоимось – float

15. КОРАБЛЬ

имя – string

водоизмещение – int

тип – string

16. СТУДЕНТ

имя – string

курс – int

пол – int(bool)

17. ИЗДЕЛИЕ

имя – string

шифр – string

количество – int

18. АДРЕС

имя – string

улица – string

номер дома – int

19. ЦЕХ

имя – string

начальник – string

количество

работающих – int

20. СТРАНА

имя – string

форма

правления – string

площадь – float

21. СЛУЖАЩИЙ

имя – string

возраст – int

рабочий стаж – int

22. БИБЛИОТЕКА

имя – string

автор – string

стоимость – float

23. ТОВАР

имя – string

количество – int

стоимость – float

24. ПЕРСОНА

имя – string

возраст – int

пол – int(bool)

25. ЖИВОТНОЕ

имя – string

класс – string

средний вес – int

26. КАДРЫ

имя – string

номер цеха – int

разряд – int

27. ЭКЗАМЕН

имя студента – string

дата – int

оценка – int

28. КВИТАНЦИЯ

номер – int

дата – int

сумма – float

29. АВТОМОБИЛЬ

марка – string

мощность – int

стоимось – float

30. КОРАБЛЬ

имя – string

водоизмещение – int

тип – string

31. ЖИВОТНОЕ

имя – string

класс – string

средний вес – int