

## Паспорт практических работ

по дисциплине «Объектно-ориентированные технологии разработки программного обеспечения», 2 семестр

### 1. Методика оценки

#### ПРАКТИЧЕСКАЯ РАБОТА №1

Задание 1. Изучить средства стандартной библиотеки C++, связанные с методами моделирования псевдослучайных чисел и вычисления выборочных характеристик.

Задание 2. Разработать программу

- генерации последовательностей случайных чисел с заданными распределениями (с одним из рекомендуемых значений параметра формы),
- вычисления выборочных характеристик: среднего арифметического, медианы, дисперсии, коэффициентов асимметрии и эксцесса, а также минимального и максимального значений.

При разработке программы удобно использовать средства стандартной библиотеки C++. Например, для хранения последовательности можно использовать подходящий контейнер, для суммирования – алгоритм accumulate и т.п.

Задание 3. Сгенерировать несколько выборок разного объема (от нескольких сотен до нескольких тысяч элементов). Вычислить по ним выборочные характеристики. Сравнить полученные выборочные характеристики с теоретическими значениями.

Задание 4. Оформить отчет по лабораторной работе. Отчет должен содержать постановку задачи, алгоритм, описание и текст разработанной программы, результаты вычислений и выводы.

Задание 5. Защитить лабораторную работу, ответив на вопросы преподавателя.

Контрольные вопросы

1. Понятие механизма случайных чисел.
2. Понятие распределения.
3. Операции с механизмами случайных чисел.
4. Выбор начального числа механизма случайных чисел: использование функции time() и класса random\_device.
5. Операции с распределениями.

#### ПРАКТИЧЕСКАЯ РАБОТА №2

Задание 1. Изучить средства стандартной библиотеки C++, связанные с потоковым вводом-выводом.

Задание 2. Разработать в соответствии с вариантом класс, в котором определить

2.1. Set- и get-функции для атрибутов класса.

2.2. Не менее одного пользовательского флага формата для управления форматом ввода-вывода и манипуляторы для управления этими флагами.

Для этого разработать некоторые форматы данных, например, форматы вывода «в файл» и «на экран» или один формат с альтернативой «без формата» (вывод всех элементов один за другим).

2.3. Пользовательские операции ввода-вывода (как перегрузку операций << и >>).

При разработке операций ввода-вывода использовать «Соглашения по созданию пользовательских операций ввода-вывода». Основной нюанс при вводе: операция ввода должна либо завершаться успешно, либо не иметь последствий, так, при несоответствии формату, ошибке ввода хотя бы одного атрибута или недопустимости его значения (например, дата 32.15.3020) требуется перевод потока в состояние ошибки (установить флаг failbit). Основной нюанс при выводе: корректный учет заданной пользователем ширины поля вывода.

Задание 3. Разработать программу, демонстрирующую функциональность созданного класса (при наличии диалога с пользователем, он должен быть вынесен за пределы класса):

- 3.1. Ввод из файла в массив (или один следующих контейнеров STL: vector, deque) и вывод в файл коллекции объектов класса.
- 3.2. Добавление отдельных объектов из отдельных файлов (или с клавиатуры).
- 3.3. Модификация объектов в контейнере (возможно, в соответствии с некоторыми критериями).
- 3.4. Удаление из коллекции объектов по некоторому признаку.
- 3.5. Поиск и вывод в отдельный файл (или на экран) объектов, удовлетворяющих некоторым условиям (примеры условий приведены в варианте).

Задание 4. Оформить отчет. Отчет должен содержать постановку задачи, алгоритм, описание и текст разработанной программы, результаты тестирования (со скриншотами) и выводы.

Задание 5. Защитить работу, ответив на вопросы преподавателя.

Контрольные вопросы

1. Стандартные потоки.
2. Стандартные потоковые операции << и >>.
3. Файловые потоки.
4. Строковые потоки.
5. Флаги и форматирующие методы.
6. Манипуляторы.
7. Вывод вещественных чисел.
8. Методы обмена с потоками.
9. Состояние потока.
10. Потоки и типы, определенные пользователем.
11. Как работают манипуляторы, пользовательские манипуляторы.
12. Пользовательские флаги форматов.
13. Итераторы потоков.

### ПРАКТИЧЕСКАЯ РАБОТА №3

Задание 1. Изучить ассоциативные контейнеры стандартной библиотеки шаблонов C++: множества, мультимножества, отображения, мультиотображения, неупорядоченные множества, неупорядоченные мультимножества, неупорядоченные отображения, неупорядоченные мультиотображения. Изучить функциональные объекты, функциональные адаптеры, лямбда-выражения, алгоритмы стандартной библиотеки шаблонов C++.

Задание 2. Переработать программу из работы №2 так, чтобы объекты разработанного класса хранились в неупорядоченном множестве или неупорядоченном мультимножестве. Использовать алгоритмы STL и/или функции-члены контейнера для реализации следующих возможностей:

- 2.1. Добавление объектов в контейнер.
- 2.2. Модификация объектов в контейнере (возможно, в соответствии с некоторыми критериями).
- 2.3. Поиск объектов в контейнере по различным критериям.

2.4. Удаление объектов из контейнера по различным критериям.

2.5. Вывод всех объектов контейнера в файл и/или на экран в отсортированном по какому-либо критерию виде (для сортировки использовать какой-нибудь подходящий вспомогательный контейнер, но не `list` и `forward_list`).

2.6. Распечатка внутренней структуры контейнера с использованием интерфейса сегментов. В программе произвести распечатку как до, так и после повторного хеширования, вызываемого принудительно или автоматически (с использованием установки подходящих значений для минимального количества сегментов и/или максимального коэффициента заполнения и добавления объектов в контейнер), и приводящего к **изменению числа сегментов**. См. пример в лекциях.

Задание 3. Для обеспечения возможности создания контейнера разработать хеш-функцию и соответствующий ей критерий эквивалентности.

Задание 4. При формировании критериев поиска, модификации, удаления использовать функциональные объекты и/или лямбда-выражения.

Задание 5. Оформить отчет. Отчет должен содержать постановку задачи, алгоритм, описание и текст разработанной программы, результаты тестирования (со скриншотами) и выводы.

Задание 6. Защитить работу, ответив на вопросы преподавателя.

Контрольные вопросы

1. Множества и мультимножества: общая характеристика, основные операции.
2. Задание критерия сортировки в множествах и мультимножествах. Задание критерия сортировки во время выполнения программы.
3. Отображения и мультиотображения: общая характеристика, основные операции.
4. Специфика доступа к элементам отображения и мультиотображения. Прямой доступ к элементам отображения (отображения как ассоциативные массивы).
5. Неупорядоченные ассоциативные контейнеры: общая характеристика, основные операции.
6. Неупорядоченные ассоциативные контейнеры: возможности создания хеш-функции и критерия эквивалентности для пользовательских типов.
7. Неупорядоченные ассоциативные контейнеры: внутреннее устройство и интерфейс сегментов.
8. Алгоритмы поиска объекта.
9. Алгоритмы, осуществляющие только чтение.
10. Алгоритмы бинарного поиска.
11. Алгоритмы записи в элементы контейнера.
12. Алгоритмы разделения и сортировки.
13. Общие функции изменения порядка.
14. Алгоритмы перестановки.
15. Алгоритмы работы с упорядоченными последовательностями как с множествами. Алгоритмы работы с пирамидами.
16. Минимальные и максимальные значения. Численные алгоритмы.
17. Концепция функциональных объектов.
18. Стандартные функциональные объекты и функциональные адаптеры.
19. Лямбда-выражения.

## 2. Критерии оценки

- Практическая работа считается **не выполненной**, если студент выполнил задание частично, отчет оформил с недостатками, ответил на вопросы слишком неполно или допустил существенные ошибки, показал недостаточное владение теоретическими и практическими навыками, оценка составляет 0–13 баллов для практической работы №1 и 0–12 баллов для остальных работ;

- Практическая работа считается выполненной **на пороговом** уровне, если студент выполнил задание с недостатками, отчет оформил с недостатками, частично ответил на вопросы, показал некоторое владение теоретическими и практическими навыками, оценка составляет 14–17 баллов для практической работы №1 и 13–16 баллов для остальных работ;
- Практическая работа считается выполненной **на базовом** уровне, если студент выполнил задание полностью, на вопросы не дал развернутых ответов, показал неполное владение основными концепциями и понятиями объектно-ориентированного подхода, отчет оформил без недостатков, оценка составляет 18–22 баллов для практической работы №1 и 17–21 баллов для остальных работ;
- Практическая работа считается выполненной **на продвинутом** уровне, если студент выполнил задание полностью, отчет оформил без недостатков, дал развернутые ответы на вопросы, показал свободное практическое владение основными концепциями и понятиями объектно-ориентированного подхода, оценка составляет 23–28 баллов для практической работы №1 и 22–26 баллов для остальных работ

### 3. Шкала оценки

В общей оценке по дисциплине баллы за практические работы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

#### 4. Перечни вариантов

##### ПРАКТИЧЕСКАЯ РАБОТА №1

1. Обобщенное гауссовское распределение, рекомендуемые значения параметра  $v$ : 1.7; 3; 5.
2. Распределение Пирсона типа II, рекомендуемые значения параметра  $v$ : 0.5; 4; 6.
3. Косинусное распределение, рекомендуемые значения параметра  $v$ : 1; 2; 6.
4. Косинусно-экспоненциальное распределение, рекомендуемые значения параметра  $v$ : 0.05; 0.1; 0.2.
5. Слэш-распределение, рекомендуемые значения параметра  $v$ : 5; 7; 10.
6. Модифицированное слэш-распределение, рекомендуемые значения параметра  $v$ : 7; 10; 15.
7. Парето-нормальное распределение, рекомендуемые значения параметра  $v$ : 5; 5.5; 6.
8. Модулируемое нормальное распределение типа I, рекомендуемые значения параметра  $v$ : 2; 3; 4.
9. Распределение Хьюбера, рекомендуемые значения параметра  $v$ : 0.001; 0.005; 0.01.

##### ПРАКТИЧЕСКАЯ РАБОТА №2

1. Класс с именем STUDENT, содержащий следующие атрибуты:

- фамилия;
- имя;
- отчество;
- название группы;
- средний балл.

Находить студентов,

- заданной группы;
- имеющих средний балл не ниже заданного.

2. Класс с именем AIRPLANE, содержащий следующие атрибуты:

- название пункта назначения;

- номер рейса;
- время вылета;
- тип самолета.

Находить самолеты, отправляющиеся

- в заданный пункт назначения;
- в течение часа после заданного времени.

3. Класс с именем ZAYAVKA (заявка на авиабилеты), содержащий следующие атрибуты:

- фамилия пассажира;
- имя пассажира;
- отчество пассажира;
- пункт назначения;
- желаемая дата вылета.

Находить заявки

- по заданной дате вылета;
- по заданному пункту назначения.

4. Класс с именем WORKER, содержащий следующие атрибуты:

- фамилия работника;
- имя работника;
- отчество работника;
- название занимаемой должности;
- год поступления на работу.

Находить работников, которые имеют

- стаж работы в организации, превышающий заданное значение;
- заданную должность.

5. Класс с именем TRAIN, содержащий следующие атрибуты:

- название пункта назначения;
- номер поезда (может содержать буквы и цифры);
- время отправления.

Находить поезда, которые

- отправляются в заданный промежуток времени;
- направляются в пункт с заданным названием.

6. Класс с именем MARSH, содержащий следующие атрибуты:

- номер маршрута;
- название начального пункта маршрута;
- название конечного пункта маршрута;
- длина маршрута в километрах.

Находить маршруты, которые

- начинаются в заданном пункте;
- имеют длину маршрута, не меньше заданной.

7. Класс с именем NOTE, содержащий следующие атрибуты:

- фамилия;
- имя;
- отчество;
- номер телефона;
- день рождения.

Находить информацию

- о человеке, номер телефона которого задан;
- о людях, чьи дни рождения приходятся на заданный месяц.

8. Класс с именем PRICE, содержащий следующие атрибуты:

- название товара;
- название магазина, в котором продается товар;

- стоимость товара в рублях.

Находить информацию

- о заданном товаре со стоимостью в заданном диапазоне;
- о товарах, продающихся в заданном магазине.

9. Определить класс с именем FILE, содержащий следующие атрибуты:

- имя файла;
- размер;
- дата создания;
- время создания.

Находить файлы, которые

- созданы в заданном диапазоне дат и времени;
- имеют размер, не меньше заданного.