ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ИМЕНИ. В.Ф. УТКИНА

Кафедра «Вычислительная и прикладная математика»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ
«Объектно-ориентированные языки и системы программирования»

Направление подготовки

09.03.03 «Прикладная информатика»

Направленность (профиль) подготовки

Прикладная информатика

Квалификация выпускника – бакалавр

Форма обучения – очная, заочная

Рязань

1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов и процедур, предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний обучающихся проводится в форме промежуточной аттестации – экзамена.

2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;

2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;

3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной

а) описание критериев и шкалы оценивания тестирования:

Шкала оценивания	**Критерий**
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 75 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 60 до 74%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 59%

б) описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	**Критерий**
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя.
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов.
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя.
0 баллов	выставляется студенту, который не смог ответить на вопрос

в) описание критериев и шкалы оценивания практического задания:

Шкала оценивания	**Критерий**
3 балла (эталонный уровень)	Задание решено верно
2 балла (продвинутый уровень)	Задание решено верно, но имеются технические неточности в выполнении
1 балл (пороговый уровень)	Задание решено верно, с дополнительными наводящими вопросами преподавателя

0 баллов	Задание не решено

На экзамен выносится: тестовое задание, 1 практическое задание и 1 теоретический вопрос. Студент может набрать максимум 9 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Шкала оценивания	**Критерий**	
отлично (эталонный уровень)	8 – 9 баллов	Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий и лабораторных работ.
хорошо (продвинутый уровень)	6 – 7 баллов	
удовлетворительно (пороговый уровень)	4 – 5 баллов	
неудовлетворительно	0 – 3 баллов	Студент не выполнил всех предусмотренных в течение семестра текущих заданий

3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

Контролируемые разделы (темы) дисциплины	**Код контролируемой компетенции (или её части)**	**Наименование оценочного средства**
Тема 1. Концепция ООП. Язык С++	ПК-1.1 ПК-1.2 ПК-1.3	Экзамен
Тема 2. Пространства имен. Простые классы. Абстракция. Инкапсуляция.	ПК-1.1 ПК-1.2 ПК-1.3	Экзамен
Тема 3. Перегрузка. Ссылки. Статические и константные члены класса.	ПК-1.1 ПК-1.2 ПК-1.3	Экзамен
Тема 4. Наследование. Перегрузка методов. Дружественные функции и классы.	ПК-1.1 ПК-1.2 ПК-1.3	Экзамен
Тема 5. Полиморфизм. Виртуальные функции. Абстрактные классы. Интерфейсы.	ПК-1.1 ПК-1.2 ПК-1.3	Экзамен
Тема 6. Типы связей между объектами. Контейнерные классы.	ПК-1.1 ПК-1.2 ПК-1.3	Экзамен
Тема 7. Множественное наследование. Шаблоны. Исключения.	ПК-1.1 ПК-1.2 ПК-1.3	Экзамен
Тема 8. STL - библиотека шаблонов. Строковые классы.	ПК-1.1 ПК-1.2 ПК-1.3	Экзамен

Для заочной формы обучения дополнительно предусмотрены контрольные работы, включающие все контролируемые разделы (темы) дисциплины.

4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1 Промежуточная аттестация (экзамен)

ПК-1: Способен разрабатывать требования, проектировать и выполнять программную реализацию программного обеспечения
ПК-1.1. Анализирует требования к программному обеспечению
ПК-1.2. Разрабатывает технические спецификации на программные компоненты
ПК-1.3. Проектирует программное обеспечение и выполняет его программную реализацию

а) типовые тестовые вопросы закрытого типа

1. Какие три основных принципа ООП?

абстракция, агрегация, ассоциация

когерентность, абстрагирование, шаблонность

переиспользование, функциональность, декларативность

инкапсуляция, наследование, полиморфизм

2. Пространства имен namespace позволяют:

объявлять новые члены класса

избежать конфликта имен

подключать новые библиотеки

указать компилятору какие имена нужно исключить на этапе компиляции

3. Стандартными потоками ввода\вывода в C++ являются:

printf и scanf

iostream и stdio

cin и cout

puts и gets

4. Уровень доступа к полям класса задаётся с помощью этих уровней доступа:

personal, defended, common

sealed, readonly, anywhere

private, protected, public

individual, covered, public

5. Деструктор класса объявляется как:

$ClassName();

~ClassName(ClassName* this);

~ClassName();

destructor ClassName(ClassName* this);

6. nullptr это:

псевдоним значения NULL

нужен для предотвращения приведения типа нулевого указателя к числу 0

присвоение к нему удаляет объект

предотвращает запись в указатель, позволяет только читать данные

7. Зачем нужен конструктор копии:

позволяет задать свой алгоритм копирования объекта, если по-умолчанию не подходит

если его не задать, объект нельзя будет скопировать

позволяет включить для объекта автоматическую сборку мусора

позволяет сделать ссылку, два объекта будут ссылаться на одни и те же данные

8. Статические поля класса позволяют:

Задать неизменяемые поля класса

Задать поля класса, которые доступны из всех экземпляров

Задать поля класса, доступные только из константных методов

Задать поля класса со статическим управлением памятью

9. Какие операторы нельзя перегружать?

?: ++ . sizeof

>> :: . sizeof

*?: :: * sizeof*

?: :: . sizeof

10. класс, производящий наследование в потомках, называют:

надкласс

подкласс

производный класс

абстрактный класс

11. Ключевое слово auto нужно для:

автоматического управления памятью с помощью сборщика мусора

автоматического вывода типа компилятором

автоматического вычисления полей-геттеров класса

автоматического выделения памяти

12. Как правильно удалить массив array[] с помощью оператора delete:

delete array;

delete array[];

delete [] array;

delete &array;

13. Истинный (параметрический) полиморфизм в C++ реализуется в виде:

шаблонов

параметры методов

полиморфизма подтипов

наследования

14. Специальный (ad-hoc) полиморфизм в C++ реализуется в виде:

шаблонов

параметры методов

полиморфизма подтипов

наследования

15. Что такое виртуальная функция?

при вызове выполняет наиболее «дочернюю» реализацию

функция не имеющая тела в данном классе

функция наследуемая от другого класса

функция встраиваемая по месту вызова

16. Какой класс называют полиморфным?

класс, не имеющий полей с данными

класс, созданный с помощью оператора new

класс, содержащий хотя бы одну виртуальную функцию

класс, не имеющий дочерних классов

16. Ключевое слово override в методах позволяет:

подтвердить совпадение сигнатур методов, и, следовательно, избежать ошибок, связанных с виртуальным переопределением метода

перегружать методы класса

объявлять методы, тела которых объявлены в дочерних классах

отказаться от виртуального переопределения

17. Интерфейсный класс это:

класс, содержащий только статические поля и методы

класс, реализующий графический интерфейс пользователя

класс, предоставляющий доступ к графическому выводу на экран компьютера

класс, который не имеет полей (переменных-членов), и все методы которого являются чисто виртуальными

18. Какие бывают типы связей между объектами классов:

перегрузка, полиморфизм, компонентность, агрегация

добавление, композиция, включение, ссылка

включение, ассоциация, зависимость, наследование

зависимость, тождественность, ссылка, наследование

19. Тип связи включение подразделяется на:

композицию и ассоциацию

композицию и агрегацию

наследование и добавление

агрегацию и зависимость

20. Контейнерные классы подразделяются на:

контейнеры-значения и контейнеры-ссылки

контейнеры-значения и контейнеры-литералы

контейнеры-ссылки и контейнеры-указатели

контейнеры-переменные и контейнеры-константы

б) типовые тестовые вопросы открытого типа

1. Что такое ООП? (Объектно-ориентированное программирование (ООП) — методология программирования, основанная на представлении программы в виде совокупности взаимодействующих друг с другом объектов, каждый из которых является экземпляром определённого класса, которые, образуют свою иерархию наследования)

2. Чем отличается объект от класса? (Объект — это некоторая сущность, обладающая определённым состоянием и поведением, имеющая определённые свойства (поля) и операции над ними (методы). Является экземпляром своего класса. Класс — представляет собой некоторый общий «шаблон» для создания объектов, инициализацию полей и описания поведения – своих методов. Фактически класс является типом для объекта (экземпляра класса).)

3. Что такое конструктор? (Конструктор - это специальный метод, который вызывается автоматически при создании экземпляра класса. При этом память под него уже выделена заранее. Позволяет выполнить сложный код для инициализации полей. Если в классе не определен явно – будет сгенерирован по-умолчанию.)

4. Что такое деструктор? (Деструктор – это специальный метод, который вызывается автоматически при удалении экземпляра класса. Часто используется для освобождения памяти, динамически выделенной в конструкторе класса.)

5. В чем отличие структур и классов в C++? (В C++ структуры (struct) аналогичны классам, только по умолчанию у них все поля public, а не private.)

6. Для чего нужен this? (В любой метод класса (включая конструкторы и деструкторы) передаётся неявный указатель this указывающий на сам объект данного класса. С его помощью метод класса определяет, с каким объектом ему предстоит работать.)

7. Для чего нужно разделять класс на реализацию cpp и заголовочный файл h? (классы в C++ обычно разделяются по файлам .h и .cpp (соответственно интерфейсную часть и реализацию).В интерфейсной части пишется описание класса, его полей и заголовков методов, в реализации – тела методов. Это позволяет осуществлять раздельную компиляцию отдельных модулей и не перекомпилировать каждый раз весь проект заново).

8. Что такое перегрузка методов? (определение нескольких методов с одинаковым именем, но различными параметрами. Список параметров может отличаться порядком следования, количеством, типом. Перегрузка нужна для того, чтобы избежать дублирования имён методов, выполняющих сходные действия, но с различной реализацией.)

9. Что такое ссылки? (Ссылка — это тип переменной в языке C++, который работает как псевдоним другого объекта или значения, объявляются через **&**. Должны быть инициализированы при создании, и не могут быть изменены после. В отличие от указателей, не могут быть нулевыми. Для доступа к значению по ссылке не требуется операция разыменовывания, т.к. это просто псевдоним переменной.)

10. Чем характеризуются константные объекты класса? (Объекты классов можно объявить константными. Их инициализация выполняется через конструкторы, после этого любая попытка изменить переменные-члены объекта запрещена (как напрямую, так и через методы). Константные объекты класса могут вызывать только константные методы класса - эти методы гарантируют, что не будут изменять сам объект или вызывать другие неконстантные методы)

11. Что такое наследование? (Наследование — это механизм создания нового класса на основе уже существующего. К существующему классу могут быть добавлены новые поля и методы, либо расширены уже существующие методы. Позволяет повторно использовать функционал уже существующих классов, когда создаваемый класс является вариантом уже существующего.)

12. Для чего нужны операторы new и delete? (C++, в отличии от C, содержит новые два оператора - new и delete, выполняющих работу с динамической памятью более эффективно и просто. Преимущества new\delete: Автоматическое вычисление размера необходимой памяти, Не надо использовать преобразование типа указателя, new не только выделяет память, но и вызывает конструкторы\деструкторы объектов, Можно перегрузить)

13. Что такое дружественный класс? (Дружественный класс – класс, имеющий доступ к приватным полям другого класса.)

14. Что такое полиморфизм? (Полиморфизм – это семейство механизмов языка программирования, позволяющее работать одному коду с различными типами, либо иметь одинаковый интерфейс доступа для различных реализаций.)

15. Что такое полиморфизм подтипов? (Полиморфизм подтипов означает, что если функция принимает некий базовый тип, то она должна также работать и со всеми его подтипами, её поведение сохранит логику и идентичность. Действительный тип объекта может при этом быть скрыт как «чёрный ящик», и предоставляться лишь по запросу идентификации объекта.)

16. Зачем нужен виртуальный деструктор? (Деструктор полиморфного базового класса должен **всегда** объявляться виртуальным. Только так обеспечивается корректное разрушение объекта производного класса через указатель на соответствующий базовый класс.)

17. Зачем нужно ключевое слово final в объявлении классов? (**final** используется в объявлении класса после его имени для запрещения наследования)

18. Что такое чисто виртуальная функция? (C++ позволяет создавать т.н. чистые виртуальные функции (или «абстрактные функции»), которые не имеют определения в родительском классе, и их обязательно переопределяют дочерние классы. Для её объявления достаточно заголовок приравнять к нулю)

18. Что такое абстрактный класс? (Абстрактный класс – класс, содержащий хотя бы одну чистую виртуальную функцию. Экземпляры абстрактного класса создавать нельзя.)

19. Что такое тип связи композиция? (Композиция (включение по значению) - включаемый объект может существовать только как часть контейнера. Если контейнер будет уничтожен, то и включённый объект тоже будет уничтожен. Объект ничего не знает о контейнере.)

20. Что такое тип связи агрегация? (Агрегация (включение по ссылке) - отношение между двумя равноправными объектами, когда объект-контейнер имеет только ссылку на другой объект. Оба объекта могут существовать независимо: если контейнер будет уничтожен, то его содержимое — нет. Но объекты по прежнему не знают о существовании контейнера.)

21. Приведите пример объявления копирующего конструктора

```
ClassName(const ClassName& other)
{
}
```

22. Приведите пример объявления класса со статическими полями и методами:

```
class Object
{
	static int field1;
	static float field2;
public:
	static void method()
	{
		//....
	}
};
```

23. Приведите пример создания и удаления экземпляра класса в динамической памяти в C++

```
Class* object = new Class();
// ...
delete object;
object = nullptr;
```

24. Приведите пример перегрузки оператора [] для класса

```
class Class
{
public:
	int operator[] (int i) const
	{
```

```
                // ....
        }
};
```

25. Приведите пример объявления шаблона класса

```
template <typename T>
class MyClass
{
        T* elems;
};
```

4.2 Типовые контрольные вопросы к экзамену

1. Что такое объектно-ориентированное программирование? Какова цель внедрения ООП?
2. Что такое объект и что такое класс? Дайте определение принципа ООП абстракции и инкапсуляции.
3. Что такое объект и что такое класс? Дайте определение принципа ООП наследования и полиморфизма.
4. Какие достоинства и недостатки ООП вы знаете?
5. Дайте краткие сведения о языке C++. Чем он отличается от C?
6. Для чего нужны пространства имен и как их объявлять? Для чего нужны using-объявления и как их
использовать?
7. Что такое потоки ввода-вывода и как с ними работать в C++?
8. Какие уровни доступа есть при объявлении класса? Формат объявления класса.
9. Что такое конструктор и деструктор? Формат их объявления.
10. Что такое указатель this и в каких случаях он применяется?
11. Зачем нужна интерфейсная часть и часть реализации при объявлении класса в программе?
12. Какие рекомендации существуют по объявлению классов?
13. Что такое юнит-тестирование? На какие части обычно разделен код теста?
14. Что такое перегрузка функций? Как компилятор определяет какая функция вызывается? Зачем
был введен nullptr и в чём его отличие от NULL?
15. Что такое конструктор копирования? В каких случаях он объявляется явно?
16. Что такое ссылки и в чём их отличие от указателей? В каких случаях надо применять ссылки, а в
каких указатели?
17. Зачем используются ссылки как параметры функции? А в качестве возвращаемого значения?
18. Что такое статические поля и методы класса? Зачем они используются?
19. Что такое перегрузка операторов и какие её виды бывают? Как она объявляется? Какие операторы нельзя перегружать?
20. Что такое константные методы класса и для чего они нужны?
21. Что такое наследование и для чего оно нужно? Что такое суперкласс? Подкласс? Базовый класс?
22. В каком порядке при наследовании вызываются конструкторы и деструкторы? Какие типы
наследования бывают?
23. Что такое переопределение метода и зачем оно используется? Зачем нужно ключевое слово
auto?

24. С помощью каких операторов происходит выделение\освобождение динамической памяти в
C++? В чём их преимущество перед функциями из C?
25. Что такое дружественная функция и класс? Когда их стоит использовать?
26. Что такое полиморфизм? Какие виды полиморфизма бывают?
27. Что такое полиморфизм? Что такое полиморфизм подтипов? Какой класс называют полиморфным?
28. Что такое виртуальная функция? Как добавить определение по умолчанию для виртуальной
функции?
29. Что такое виртуальная функция? В чём отличие виртуальных функций от обычного переопределения?
30. Для чего используется виртуальный деструктор? Для чего нужны модификаторы override и final?
31. Что такое чистые виртуальные функции? Что такое абстрактный класс и для чего он используется?
32. Что такое интерфейсный класс и для чего он используется?
33. Что такое тип связи композиция? Приведите пример.
34. Что такое тип связи агрегация? Приведите пример.
35. Что такое тип связи ассоциация? Приведите пример.36. Что такое тип связи зависимость? Приведите пример.
37. Что такое тип связи наследование? Приведите пример.
38. Что такое контейнерные классы и какие виды их бывают?
39. Что такое множественное наследование? Что такое ромбовидное наследование и какие проблемы с этим связаны?
40. Что такое множественное наследование? Что такое виртуальное наследование и для чего оно
применяется в языке C++?
41. Что такое шаблоны и для чего они применяются? Как объявить шаблон функции и шаблон класса?
42. Что такое инстанцирование шаблона и в какой момент оно происходит? Какие преимущества и
недостатки есть у шаблонов?
43. Что такое non-type параметры шаблона, какого типа они могут быть? Что такое явная специализация шаблона и каким образом она объявляется?
44. Что такое частичная специализация шаблона и каким образом она объявляется? Какое ограничение существует у частичной специализации шаблона и каким образом его можно обходить?
45. Чем характеризуется приведение типов с помощью static_cast и dynamic_cast?
46. Чем характеризуется приведение типов с помощью const_cast и reinterpret_cast?
47. Что такое enum class и в чём его отличие от обычного enum?
48. Что такое исключения? В чем преимущество исключений перед возвратом кодов ошибок?
49. Что такое исключения? Какова общая структура выбрасывания и обработки исключения? Из каких
блоков она состоит?
50. Что такое исключения? Для чего нужны классы-исключений? Каковы недостатки исключений?
51. Что такое исключения? В каких случаях стоит использовать исключения?
52. Что такое STL и из каких частей она состоит? Что такое последовательные контейнеры?

53. Что такое STL и из каких частей она состоит? Какие есть стандартные последовательные
контейнеры в STL?
54. Что такое итераторы и для чего они применяются?
55. Какие перегруженные операторы имеют итераторы?
56. Какие есть основные методы для работы с итераторами?
57. Какие типы итераторов обычно предоставляют контейнеры?
58. Что такое STL и из каких частей она состоит? Что такое ассоциативные контейнеры?
59. Что такое STL и из каких частей она состоит? Какие есть стандартные ассоциативные контейнеры в
STL?
60. Что такое адаптеры? Какие есть стандартные адаптеры в STL?
61. Что такое алгоритмы STL и как они реализованы?
62. Что такое объект std::initializer_list и для чего он используется?
63. Что такое объект std::string и какие основные методы имеет?
64. Что такое идиома RAII?
65. Что такое умный указатель и для чего он используется?
66. Какие рекомендуемые умные указатели есть в стандартной библиотеке?
67. Что такое циклическая зависимость, к каким проблемам она может приводить и какие пути
решения существуют?
68. Что такое лямбда-выражения и какой формат их объявления?

4.3 Типовые задачи к экзамену по дисциплине

1. Описать массив классов и поместить в него сгенерированные сведения о N книгах. Предусмотреть такие сведения как название книги, жанр, дата издания (отдельный объект), количество экземпляров, ФИО автора (отдельным объектом), количество страниц. Написать функцию выдачи списка книг по фамилии автора, жанру или диапазону годов издания. Написать функцию удаления сведений о количестве страниц, если количество страниц менее заданного числа. Написать функцию добавления информации о возрасте книги, найденую по дате её издания.
2. Описать массив классов и поместить в него сгенерированные сведения о N работниках. Предусмотреть такие сведения как ФИО работника (отдельным объектом), дата рождения (отдельный объект), номер телефона, место работы (отдельный объект со сведениями о названии организации, должности и стаже). Написать функцию выдачи списка работников по названию организации, должности или диапазону стажа. Написать функцию удаления сведений о дате рождения, если стаж менее заданного числа. Написать функцию добавления информации о районе проживания работника, найденного по первым двум цифрам телефона.
3. Описать массив классов и поместить в него сгенерированные сведения о N студентах. Предусмотреть такие сведения как ФИО студента (отдельным объектом), дата поступления (отдельный объект), номер телефона, результаты сессии (отдельный массив с информацией о названии предметов и полученных оценках). Написать функцию удаления сведений о дате поступления, если год поступления старше заданного. Написать функцию добавления информации о среднем балле студента, найденного по оценкам сессии. Написать функцию выдачи списка студентов отсортированному по убыванию среднего балла.
4. Описать массив классов и поместить в него сгенерированные сведения о N деталях. Предусмотреть такие сведения как наименование детали, габаритные размеры (отдельный объект), материал, масса детали, список поставщиков деталей (отдельный массив из названий организации и контактного телефона). Написать функцию удаления

сведений о материале, если масса детали менее указанной величины. Написать функцию добавления информации о габаритном объёме детали, найденного по габаритным размерам. Написать функцию выдачи списка деталей отсортированному по убыванию массы.

5. Описать массив классов и поместить в него сгенерированные сведения о N сданных экзаменационных сессий. Предусмотреть такие сведения сессии как номер курса, дата начала сессии (отдельный объект), дата конца сессии (отдельный объект), список предметов (массив со сведениями о названии предмета и полученной оценки). Написать функцию удаления сведений о номере курса, если номер сессии нечётный. Написать функцию добавления информации о средней оценки сессии, найденного по списку оценок предметов. Написать функцию выдачи списка предметов и оценок лучшей сессии и худшей сессии.

6. Описать массив классов и поместить в него сгенерированные сведения о N людях. Предусмотреть такие сведения как ФИО человека (отдельным объектом), пол, дата рождения (отдельный объект), номер телефона, адрес проживания (отдельный объект содержащий сведения о городе, улице, номере дома и номере квартиры). Написать функцию выдачи списка людей по городу, полу или диапазону годов рождения. Написать функцию удаления сведений о дате рождения, если год рождения более указанного. Написать функцию добавления информации о районе проживания работника, найденного по первым двум цифрам телефона.

7. Описать массив классов и поместить в него сгенерированные сведения о N работниках. Предусмотреть такие сведения как ФИО работника (отдельным объектом), дата рождения (отдельный объект), номер цеха, трудовая информация (отдельный объект со сведениями о должности, разряде, стаже). Написать функцию выдачи списка работников по должности, разряду или диапазону стажа. Написать функцию удаления сведений о дате рождения, если стаж менее заданного числа. Написать функцию добавления информации о возрасте работника, найденного по году рождения.

8. Описать массив классов и поместить в него сгенерированные сведения о N сотрудниках. Предусмотреть такие сведения как ФИО сотрудника (отдельным объектом), дата рождения (отдельный объект), должность, стаж, зарплата (отдельный объект со сведениями о окладе, премии, оплате интенсивности, оплате переработки). Написать функцию удаления сведений о дате рождения, если стаж менее заданного числа. Написать функцию добавления информации о суммарном доходе работника, найденного как сумма всей составляющей зарплаты минус 13%. Написать функцию выдачи списка работников, отсортированных по убыванию дохода.

9. Описать массив классов и поместить в него сгенерированные сведения о плане выпуска N наименований. Предусмотреть такие сведения как название изделия, шифр, единица измерения, план выпуска (отдельный объект из плана выпуска и сколько фактически выпущено), список заказчиков (отдельный массив из названий организации и количества закупаемого наименования). Написать функцию удаления сведений о единице измерения, если план выпуска менее заданного числа. Написать функцию добавления информации о проценте выполнения плана, найденного как соотношение фактического выпуска от плана выпуска. Написать функцию выдачи списка изделий, с перевыполнением плана, списка изделий с недовыполнением плана.

10. Описать массив классов и поместить в него сгенерированные сведения о N спортсменах. Предусмотреть такие сведения как ФИО спортсмена (отдельным объектом), дата рождения (отдельный объект), страна, вид соревнования, результаты соревнований (отдельный объект со сведениями о названии соревнования, дате проведения, результате спортсмена). Написать функцию выдачи списка спортсменов по названию соревнования, стране или диапазону годов рождения. Написать функцию удаления сведений о дате

рождения, если год рождения менее заданного числа. Написать функцию добавления информации о среднем результате спортсмена по всем соревнованиям.

11. Описать массив классов и поместить в него сгенерированные сведения о N футболистах. Предусмотреть такие сведения как ФИО футболиста (отдельным объектом), дата рождения (отдельный объект), количество голов, команда (отдельный объект со сведениями о названии команды, стране, дате вступления (отдельный объект), зарплате футболиста). Написать функцию выдачи списка футболистов по названию команды, стране или диапазону забитых голов. Написать функцию удаления сведений о дате рождения, если год рождения менее заданного числа. Написать функцию добавления информации о количестве лет нахождения в команде, рассчитанной по году вступления в команду.

12. Описать массив классов и поместить в него сгенерированные сведения о инвентаризационной ведомости из N наименований. Предусмотреть такие сведения как название наименования, инвентарный номер, дата принятия на учёт (отдельный объект), количество, место хранения (отдельный объект из полей номер корпуса, номер этажа, номер помещения). Написать функцию удаления сведений о дате принятия на учёт, если год принятия является текущим. Написать функцию добавления информации о сроке службы наименования по текущей дате и дате принятия его на учёт. Написать функцию выдачи списка наименований по номеру корпуса, номеру этажа или с указанным диапазоном сроков службы.

13. «Комплексное число» – Complex. Разработать класс комплексных чисел. Класс должен работать с функциями для изменения и получения значения действительной и мнимой части, для реализации операций сложения, вычитания, умножения, деления, присваивания комплексных чисел. Предусмотреть функцию toString. Создать 2 массива классов и с помощью них поэлементно показать работу всех операций.

14. «Дробь» – Fraction. Разработать класс в виде пары целых положительных чисел (m,n) а также отдельно знак дроби. Класс должен работать с функциями для изменения и получения значения числителя и знаменателя, сложения, вычитания, умножения, деления и присваивания дробей. Предусмотреть функцию toString. Создать 2 массива классов и с помощью них поэлементно показать работу всех операций.

15. «Вектор» – Vector. Разработать класс вектора размерности n. Реализовать функции для изменения и получения значения компонента вектора, вычисления длины вектора, скалярного произведения, сложения, умножения, умножения на скаляр. Предусмотреть функцию toString. Создать 2 массива классов и с помощью них поэлементно показать работу всех операций.

16. «Квадратная матрица» – Matrix. Разработать класс квадратной матрицы n x n. Реализовать функции для изменения и получения значения элемента матрицы, сложения, вычитания, умножения матриц; вычисления индексов максимального и минимального элемента матрицы. Предусмотреть функцию toString. Создать 2 массива классов и с помощью них поэлементно показать работу всех операций.

17. «Многочлен» – Polynom. Разработать класс полинома степени n. Реализовать функции для изменения и получения значения указанного коэффициента, вычисления значения полинома; сложения, вычитания, умножения полиномов. Предусмотреть функцию toString. Создать 2 массива классов и с помощью них поэлементно показать работу всех операций.

18. «Фигуры» – Shapes. Разработать класс для описания плоских фигур: круг, прямоугольник, треугольник. Включить функции для получения и изменения параметров фигур, перемещения на плоскости, вращения, нахождения площади и периметра фигуры. Предусмотреть функцию toString. Выполнить тестирование модуля, создав массив и показав на его примере работу всех функций.

19. «Множество целых чисел» –Set. Разработать класс множества целых чисел мощности n. Реализовать функции для определения принадлежности заданного элемента множеству, добавление\удаление элемента, пересечения, объединения, разности двух множеств. Предусмотреть функцию toString. Создать 2 массива классов и с помощью них поэлементно показать работу всех операций.

20. «Массив строк» – StringArray. Разработать класс для представления массива строк. Реализовать функции для добавления\удаления строк, для поэлементной конкатенации двух массивов, упорядочения строк по длине, слияния двух массивов строк с удалением повторяющихся строк, а также формирование массива количества слов в каждой строке. Предусмотреть функцию toString. Создать 2 класса и с помощью них поэлементно показать работу всех функций.

21. «Массив бит» – BitArray. Разработать класс представляющий собой массив битов длины n. Реализовать функции для установки и получения значения бита на заданной позиции, изменения размера массива (справа и слева), сдвиг битов вправо\влево на заданное число позиций, битовые операции and и or для двух массивов. Предусмотреть функцию toString. Создать 2 массива классов и с помощью них поэлементно показать работу всех операций.

22. «Булева матрица» – BoolMatrix. Разработать класс представляющий собой матрицу булевых значений размерности n×m. Реализовать функции для изменения и получения значения указанного элемента, логического сложения, умножения и инверсии матриц. Реализовать функцию для подсчета количества true и false значений в матрице. Предусмотреть функцию toString. Создать 2 массива классов и с помощью них поэлементно показать работу всех операций.

23. «Односвязный список» – LinkedList. Разработать класс для работы с односвязным списком с целыми числами. Реализовать функции добавления элемента на заданную позицию, удаление всех элементов с заданным значением, получение значения по заданному индексу, объединение двух списков, разделение списка на два с указанной позиции, реверс списка. Предусмотреть функцию toString. Создать 2 массива классов и с помощью них поэлементно показать работу всех операций.

24. «Бинарное дерево» – BinaryTree. Разработать класс для работы с бинарным деревом, узлы которого содержат натуральные числа. Реализовать функции добавления и удаления узлов, получения массива узлов с заданным значением, определения высоты и количества листьев у дерева. Предусмотреть функцию toString. Создать массив и с помощью них поэлементно показать работу всех операций.