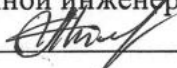


Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»

Рыбницкий филиал ПГУ
Кафедра «Информатики и программной инженерии»

««УТВЕРЖДАЮ»»

Заведующий кафедрой информатики и
программной инженерии

доцент  Л.А. Тягульская

«19» 09 2024г.

Фонд оценочных средств ПО ДИСЦИПЛИНЕ «ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Направление подготовки

09.03.04 Программная инженерия

(Код и наименование направления подготовки)

Профиль подготовки

«Разработка программно-информационных систем»

(наименование профиля подготовки)

Квалификация (степень)

Бакалавр

Форма обучения

очная

Год набора **2023**

Разработал: преподаватель

 О.М. Нагаевский

«19» 09 2024г.

Рыбница 2024г.

Паспорт фонда оценочных средств по учебной дисциплине

1. В результате изучения дисциплины «Проектирование программного обеспечения» у обучающихся должны быть сформированы следующие компетенции:

Категория (группа) компетенций	Код и наименование	Код и наименование индикатора достижения универсальной компетенции
Универсальные компетенции и индикаторы их достижения		
Системное и критическое мышление	УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИД УК-1.1. Выбирает источники информации, адекватные поставленным задачам и соответствующие научному мировоззрению ИД УК-1.2. Демонстрирует умение рассматривать различные точки зрения на поставленную задачу в рамках научного мировоззрения и определять рациональные идеи ИД УК-1.3. Выявляет степень доказательности различных точек зрения на поставленную задачу в рамках научного мировоззрения
Командная работа и лидерство	УК-3. Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	ИД УК-3.1. Знает различные приемы и способы социализации личности и социального взаимодействия. ИД УК-3.2. Умеет строить отношения с окружающими людьми, с коллегами. ИД УК-3.3. Имеет практический опыт участия в командной работе, в социальных проектах, распределения ролей в условиях командного взаимодействия.
. Общепрофессиональные компетенции выпускников и индикаторы их достижения		
	ОПК-1. Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ИД ОПК-1.1. Знает основы математики, физики, вычислительной техники и программирования. ИД ОПК-1.2. Умеет решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования. ИД ОПК-1.3. Имеет навыки теоретического и экспериментального исследования объектов профессиональной деятельности.
	ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	ИД ОПК-2.1. Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. ИД ОПК-2.2. Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. ИД ОПК-2.3. Имеет навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.
	ОПК-3. Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-	ИД ОПК-3.1. Знает принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных

	коммуникационных технологий и с учетом основных требований информационной безопасности	технологий и с учетом основных требований информационной безопасности. ИД ОПК-3.2. Умеет решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности. ИД ОПК-3.3. Имеет навыки подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций, и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.
	ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ИД ОПК-6.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий. ИД ОПК-6.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ИД ОПК-6.3. Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач.
	ОПК-7. Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	ИД ОПК-7.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий. ИД ОПК-7.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ИД ОПК-7.3. Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач.
Обязательные профессиональные компетенции выпускников и индикаторы их достижения		
	ПК-5. Владение навыками использования различных технологий разработки программного обеспечения	ИД ПК-5.1. Знает современные технологии разработки ПО (структурное, объектно-ориентированное) ИД ПК-5.2. Умеет использовать современные технологии разработки ПО ИД ПК-5.3. Имеет навыки использования современных технологий разработки ПО
	ПК-6. Владение концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества	ИД ПК-6.1. Знает концепции и атрибуты качества ПО ИД ПК-6.2. Умеет определять атрибуты качества ПО ИД ПК-6.3. Имеет навыки в использовании методов, инструментов и технологий обеспечения качества ПО

2. Программа оценивания контролируемой компетенции:

№ п\п	Контролируемые модули, разделы (темы) дисциплины и их наименование	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Основы объектно-ориентированного подхода	УК-1, УК-3, ОПК-1, ОПК-2, ОПК-3, ОПК-6, ОПК-7, ПК-5, ПК-6	Лабораторные работы. Тестирование.
2	Стандартная библиотека шаблонов STL.	УК-1, УК-3, ОПК-1, ОПК-2, ОПК-3, ОПК-6, ОПК-7, ПК-5, ПК-6	Лабораторные работы. Тестирование.
3	Паттерны проектирования.	УК-1, УК-3, ОПК-1, ОПК-2, ОПК-3, ОПК-6, ОПК-7, ПК-5, ПК-6	Лабораторные работы. Тестирование.
4	Промежуточная аттестация	УК-1, УК-3, ОПК-1, ОПК-2, ОПК-3, ОПК-6, ОПК-7, ПК-5, ПК-6	Экзамен. Защита курсовой

«УТВЕРЖДАЮ»

зав. кафедрой ИиПИ,

доцент Л. А. Тягульская

«19» 09 2024 г.

Итоговый тест по дисциплине
«Проектирование программного обеспечения»
для студентов II курса
направления «Программная инженерия» профиля подготовки
«Разработка программно-информационных систем»

ДЕ-1

1. Какой код достаточно добавить в функцию inc() вместо комментария, чтобы компиляция прошла без ошибок? (Выберите несколько вариантов ответа).

```
float inc() {  
    try { counter++;  
        if(counter>max)throw 1;  
        cout<<counter; }  
    //комментарий  
}
```

- a) catch (...){};
- b) catch (int e){};
- c) finally{};
- d) return counter;

2. Какое значение будет возвращено при вызове meth(5)?

```
int meth(int x) {  
    int y = 10;           //1  
    try { y += x;         //2  
        if(x<=5) throw new Exception(); //3  
        y++; }           //4  
    catch(Exception e) { y--; } //5  
    return y; }          //6
```

- a) 12;
- b) 13;
- c) 14;
- d) 15;

3. Какое значение будет возвращено при вызове meth(12), если при вызове mexcept(int x) возникает исключительная ситуация?

```
int meth(int x) {  
    int count=0;  
    try { count += x;  
        count += mexcept(count);  
        count++;  
    } catch(...) {  
        --count;  
    }  
    return count;  
}
```

}

- a) 11;
- b) 12;
- c) 13;
- d) 14;

4. Что такое файловый потоковый ввод-вывод?

- a) Потоковый вывод на консоль, перенаправленный в файл;
- b) Потоковый ввод с консоли, перенаправленный на чтение из файла;
- c) Файловый ввод вывод средствами операционной системы;
- d) Динамическое связывание и синхронизация буфера в операционной системе и дескриптора файла, позволяющее работать с файлами способом, аналогично работе с устройствами;

5. Что такое стандартный поток ошибок?

- a) Поток для вывода любого текста на консоль;
- b) Синоним стандартного потокового вывода;
- c) Поток для вывода диагностических сообщений на консоль, отдельно от стандартного потока вывода;
- d) Такого потока не существует

6. Как обозначается стандартный поток ошибок на языке Си?

- a) cout;
- b) stdout;
- c) stderr;
- d) cerr;

7. Как обозначается стандартный поток ошибок на языке C++?

- a) cout;
- b) stdout;
- c) stderr;
- d) cerr;

8. Какие функции языка C++ используются для открытия потока файлов и связывания его с файлом? (Выберите несколько вариантов ответа).

- a) Конструктор создания потока;
- b) метод open;
- c) метод close;
- d) метод tellg;

9. Какое из указанных действий приведет к тому, что поток, наследник класса TThread переходит в состояние "пассивный"?

- a) вызов метода Suspend ();
- b) вызов метода Pause();
- c) окончание выполнения метода Execute;
- d) вызов метода notifyAll();

10. Какая WinApi функция запускает остановленный поток?

- a) Resume;
- b) ResumeThread;
- c) Execute;
- d) StartThread;

11. Ядро STL библиотеки шаблонов состоит из следующих основных компонентов (Выберите несколько вариантов ответа).
- a) контейнеров
 - b) множеств;
 - c) аллокаторов;
 - d) итераторов;
12. Преимуществом библиотеки STL является
- a) отсутствие фиксированного стандартного представления;
 - b) доступ к итератору приводит к вызову виртуальной функции;
 - c) обеспечение стандартного интерфейса в виде набора операций;
 - d) предотвращение выхода за пределы контейнера возлагается на программиста;
13. Недостатком библиотеки STL является
- a) доступ к итератору приводит к вызову виртуальной функции;
 - b) однородность контейнеров по определению;
 - c) обеспечение стандартного интерфейса в виде набора операций;
 - d) возможность использования алгоритмов для любого контейнера
14. Контейнер это класс, предназначенный для ...
- a) хранения объектов
 - b) выделения памяти
 - c) доступа к элементам
 - d) выполнения операций поиска
15. Контейнер ... НЕ входит в библиотеку STL
- a) Multiset
 - b) Deque
 - c) Tree
 - d) Bitset
16. Группа операций, НЕ являющихся стандартной для контейнеров это операции...
- a) вставки элемента
 - b) удаления элемента
 - c) итераторов
 - d) аллокаторов
17. Аллокатор это класс, предназначенный для ...
- a) хранения объектов
 - b) выделения памяти
 - c) доступа к элементам
 - d) выполнения операций поиска
18. Стандартный распределитель памяти выделяет память при помощи операции ...
- a) delete
 - b) allocate
 - c) malloc
 - d) new
19. Итератор это класс, предназначенный для ...
- a) хранения объектов

- b) выделения памяти
- c) доступа к элементам
- d) выполнения операций поиска

20. Распространенные операции с контейнерами не реализованные методами контейнеров называются ...

- a) аллокаторами
- b) итераторами
- c) алгоритмами поиска
- d) алгоритмами STL

21. ... НЕ принадлежит к группам алгоритмов

- a) модифицирующие алгоритмы
- b) немодифицирующие алгоритмы
- c) алгоритмы поиска
- d) алгоритмы сортировки

22. Удаление всех элементов вектора происходит при помощи метода

- a) delete
- b) clear
- c) erase
- d) pop_back

23. Получение количества элементов происходит при помощи метода

- a) size
- b) length
- c) max_size
- d) capacity

24. Доступ к первому элементу вектора происходит при помощи метода

- a) front
- b) first
- c) begin
- d) rbegin

25. Доступ к первому элементу списка myList происходит при помощи ...

- a) myList[0]
- b) myList.front()
- c) myList.begin()
- d) myList.at(0)

26. Для доступа к элементам списка myList используется метод ... (Выберите несколько вариантов ответа).

- a) myList[n]
- b) myList.back()
- c) myList.front()
- d) myList.at(n)

27. Для получения итератора на первый элемент списка используется метод ...

- a) myList[0]
- b) myList.front()
- c) myList.begin()

d) myList.at(0)

ДЕ-3

28. Паттерн проектирования это ...(Выберите несколько вариантов ответа).

- a) повторяемая архитектурная конструкция
- b) абстрактный тип данных
- c) отношения и взаимодействия между классами или объектами
- d) шаблон класса стандартной библиотеки

29. Какие из перечисленных паттернов проектирования являются антипаттернами? (Выберите несколько вариантов ответа).

- a) Божественный объект
- b) Инверсия абстракции
- c) Абстрактная фабрика
- d) Null Object
- e) Фасад

30. Укажите отрицательные стороны применения паттернов проектирования: (Выберите несколько вариантов ответа).

- a) Могут использоваться там, где их применение не оправдано и не требуется
- b) Могут затруднять понимание бизнес-логики приложения
- c) Всегда усложняют программную архитектуру приложения
- d) Не предлагают конкретных алгоритмов решения задач в определенной предметной области

31. К типам паттернов относятся ... шаблоны.

- a) адаптивные
- b) интерпретируемые
- c) делегируемые
- d) поведенческие

32. Делегирование относится к шаблонам.

- a) поведенческим
- b) структурным
- c) порождающим
- d) основным

33. Шаблон «Делегирование» представляет собой

- a) объект, передающий ответственность за выполнение связанному объекту
- b) класс, представляющий интерфейс для создания компонентов системы
- c) класс, обеспечивающий простой способ доступа к другим классам
- d) класс, представляющий собой интерфейс для работы с набором

34. Целью шаблона «Делегирования» является обеспечение ...

- a) оповещения об изменениях зависящих объектов
- b) независимости конструирования объекта и его представления
- c) возможности изменить поведение конкретного экземпляра объекта
- d) независимости классов отправителя и получателя

35. В приведенном ниже коде используется шаблон ...

```
class I {  
public:
```

```
virtual void f() = 0;
virtual void g() = 0;
};
```

```
class A : public I {
public:
void f() { std::cout << "A: вызываем метод f()" << std::endl; }
void g() { std::cout << "A: вызываем метод g()" << std::endl; }
};
```

```
class B : public I {
public:
void f() { std::cout << "B: вызываем метод f()" << std::endl; }
void g() { std::cout << "B: вызываем метод g()" << std::endl; }
};
```

```
class C : public I {
public:
C() : m_i ( new A() ) { }
virtual ~C() {
delete m_i;
}
void f() { m_i->f(); }
void g() { m_i->g(); }
void toA() {
delete m_i;
m_i = new A();
}
void toB() {
delete m_i;
m_i = new B();
}
private:
I * m_i;
};
```

- a) фабричный метод
- b) делегирования
- c) прототип
- d) интерфейс

36. Минусом использования шаблона делегирования является ...

- a) затруднение оптимизации по скорости
- b) изоляция конкретных классов
- c) оповещения об изменениях зависящих объектов
- d) упрощение замены семейств продуктов

37. Какой из перечисленных шаблонов считается фундаментальным(Fundamental)?

- a) Фабричный метод
- b) Отложенная инициализация
- c) Прототип

d) Интрефейс

38. Интерфейс относится к шаблонам.

- a) поведенческим
- b) структурным
- c) порождающим
- d) основным

39. Шаблон «Интерфейс» представляет собой

- a) класс, представляющий интерфейс для создания сложного объекта
- b) класс, представляющий интерфейс для создания компонентов системы
- c) класс, обеспечивающий простой способ доступа к другим классам
- d) класс, представляющий собой интерфейс для работы с набором

40. Целью шаблона «Интерфейс» является обеспечение ...

- a) оповещения об изменениях зависящих объектов
- b) независимости конструирования объекта и его представления
- c) представления интерфейса для более простого доступа к другим классам
- d) независимости классов отправителя и получателя

41. В представленном ниже коде используется шаблон ...

```
class A
{
public:
static A * create ();
virtual ~A () {}
virtual void do_something() = 0;
};
class B : public A
{
public:
void do_something();
};
```

- a) Фабричный метод
- b) Отложенная инициализация
- c) Прототип
- d) Интрефейс

42. Какие из шаблонов относятся к порождающим? (Выберите несколько вариантов ответа).

- a) Abstract Factory;
- b) Command;
- c) Strategy;
- d) Singleton.

43. Абстрактная фабрика относится к шаблонам.

- a) поведенческим
- b) структурным
- c) порождающим
- d) основным

44. Шаблон «Абстрактная фабрика» представляет собой

- a) класс, представляющий интерфейс для создания сложного объекта
 - b) класс, представляющий интерфейс для создания компонентов системы
 - c) структура, разделить абстракцию и реализацию для независимого изменения
 - d) класс, представляющий собой интерфейс для работы с набором
45. Целью шаблона «Абстрактная фабрика» является обеспечение ...
- a) оповещения об изменениях зависящих объектов
 - b) независимости конструирования объекта и его представления
 - c) представление интерфейса для создания семейств взаимосвязанных объектов
 - d) независимости классов отправителя и получателя
46. Плюсы от применения шаблона «Абстрактная фабрика» является ... (Выберите несколько вариантов ответа)
- a) возможность изменения внутреннего представления продукта
 - b) изоляция конкретных классов
 - c) оповещения об изменениях зависящих объектов
 - d) упрощение замены семейств продуктов
47. Строитель относится к шаблонам.
- a) поведенческим
 - b) структурным
 - c) порождающим
 - d) основным
48. Шаблон «Строитель» представляет собой
- a) класс, представляющий интерфейс для создания сложного объекта
 - b) объект, инициализируемый во время первого обращения к нему
 - c) структура, разделить абстракцию и реализацию для независимого изменения
 - d) класс, представляющий собой интерфейс для работы с набором
49. Целью шаблона «Строитель» является обеспечение ...
- a) оповещения об изменениях зависящих объектов
 - b) независимости конструирования объекта и его представления
 - c) создание класса-оболочки с требуемым интерфейсом
 - d) независимости классов отправителя и получателя
50. Плюсы от применения шаблона «Строитель» является ... (Выберите несколько вариантов ответа)
- a) возможность изменения внутреннего представления продукта
 - b) изоляция кода, реализующего конструирование и представление
 - c) оповещения об изменениях зависящих объектов
 - d) независимое изменение абстракции и реализации
51. На рисунке изображен шаблон ...
- a) Адаптер
 - b) Фабричный метод
 - c) Строитель
 - d) Абстрактная фабрика
52. В чём главное различие паттернов Adapter и Proxy?
- a) Adapter предоставляет другой интерфейс к объекту, Proxy в точности повторяет интерфейс объекта

- b) Эти паттерны нельзя сравнивать
- c) Adapter создаёт обёртку вокруг одного объекта, Proxy - вокруг множества
- d) Adapter добавляет объекту новый функционал (в отличии от Proxy)

53. Применение, какого шаблона проектирования демонстрирует следующий код?

```
List<String> strings = Arrays.asList("hello, ", "world", "!");
```

- a) Decorator
- b) Adapter
- c) Abstract factory
- d) Facade
- e) Builder

54. Адаптер относится к шаблонам.

- a) поведенческим
- b) структурным
- c) порождающим
- d) основным

55. Шаблон «Адаптер» представляет собой

- a) класс, расширяющий функциональность другого класса
- b) объект, который абстрагирует работу с несколькими классами
- c) структура, разделить абстракцию и реализацию для независимого изменения
- d) использование функций объекта, недоступного для модификации

56. Целью шаблона «Адаптер» является обеспечение ...

- a) оповещения об изменениях зависящих объектов
- b) независимое изменение абстракции и реализации
- c) создание класса-оболочки с требуемым интерфейсом
- d) независимости классов отправителя и получателя

57. Мост относится к шаблонам.

- a) поведенческим
- b) структурным
- c) порождающим
- d) основным

58. Шаблон «Мост» представляет собой

- a) класс, расширяющий функциональность другого класса
- b) объект, который абстрагирует работу с несколькими классами
- c) структура, разделить абстракцию и реализацию для независимого изменения
- d) объект, который объединяет в себе объекты, подобные ему самому

59. Целью шаблона «Мост» является обеспечение ...

- a) оповещения об изменениях зависящих объектов
- b) независимое изменение абстракции и реализации
- c) взаимодействия множества объектов
- d) независимости классов отправителя и получателя

60. В нижеприведенном коде используется паттерн ...

```
class C {  
public:
```

```
virtual void f() = 0;
};
```

```
class D : public C{
public:
void f() /*...*/;
};
```

```
class E : public C {
public:
void f() /*...*/;

};
```

```
class A {
protected:
C* c;
public:
virtual void F() = 0;
};
class B {
public:
void F() { c->f(); };
};
```

- a) Decorator
- b) Adapter
- c) Facade
- d) Bridge

61. Шаблон разделяющий абстракцию и реализацию это ...

- a) Bridge
- b) Adapter
- c) Decorator
- d) Facade

62. Фасад относится к шаблонам.

- a) поведенческим
- b) структурным
- c) порождающим
- d) основным

63. Шаблон «Фасад» представляет собой

- a) класс, расширяющий функциональность другого класса
- b) объект, который абстрагирует работу с несколькими классами
- c) структура, разделить абстракцию и реализацию для независимого изменения
- d) объект, который объединяет в себе объекты, подобные ему самому

64. Целью шаблона «Фасад» является обеспечение ...

- a) оповещения об изменениях зависящих объектов
- b) независимое изменение абстракции и реализации
- c) объединения нескольких классов в единое целое

d) независимости классов отправителя и получателя

65. Шаблон «Фасад» рекомендован для использования в условиях

- a) наличия группы объектов, которые могут обрабатывать сообщения конкретного типа
- b) необходимости предоставления простого интерфейса к сложной системе
- c) необходимости изменять поведение объектов на стадии выполнения
- d) необходимости изменять поведение объектов в зависимости от состояния

66. Цепочка обязанностей относится к шаблонам.

- a) поведенческим
- b) структурным
- c) порождающим
- d) основным

67. Шаблон «Цепочка обязанностей» представляет собой ...

- a) действие и его параметры
- b) организацию в системе уровней ответственности
- c) операцию, выполняющуюся над объектами других классов
- d) цепочку объектов, связанных правилами булевой логики

68. Целью шаблона «Цепочка обязанностей» является обеспечение ...

- a) оповещения об изменениях зависящих объектов
- b) взаимодействия множества объектов
- c) связывание объектов-получателей в цепочку, позволяющую обработать запрос нескольким объектам
- d) независимости классов отправителя и получателя

69. Шаблон «Цепочка обязанностей» НЕ рекомендован для использования в условиях

- a) наличия группы объектов, которые могут обрабатывать сообщения конкретного типа
- b) необходимости обработки сообщений хотя бы одним объектом
- c) необходимости обработки сообщений по схеме «обработай сам либо перешли другому»
- d) необходимости строго указать объект, который будет обрабатывать сообщение

70. Шаблон «Цепочка обязанностей» рекомендован для использования в условиях

- a) необходимости обработки сообщений по схеме «обработай сам либо перешли другому»
- b) необходимости обеспечения различных вариантов алгоритма или поведения
- c) необходимости изменять поведение объектов на стадии выполнения
- d) необходимости изменять поведение объектов в зависимости от состояния

71. Какой шаблон проектирования из нижеперечисленных используется в следующем фрагменте кода?

```
JButton jbOk = new JButton("OK");
```

```
class ActionListener{  
public:  
void actionPerformed(ActionEvent e) {  
cout<<"OK";  
}  
};
```

```
ActionListener *actionListener=new ActionListener();
```

```
jbOk.addActionListener(actionListener);
```

```
class AbstractAction
```

```
{
```

```
public:
```

```
void actionPerformed(ActionEvent e)
```

```
{
```

```
Exit(0);
```

```
}
```

```
};
```

```
Action *cancelAction = new AbstractAction();
```

a) Состояние

b) Стратегия

c) Декоратор

d) Команда

e) Прототип

72. Команда относится к шаблонам.

a) порождающим

b) структурным

c) поведенческим

d) основным

73. Шаблон «Команда» представляет собой ...

a) действие и его параметры

b) состояние объекта другого класса

c) операцию, выполняющуюся над объектами других классов

d) цепочку объектов, связанных правилами булевой логики

74. Целью шаблона «Команда» является обеспечение ...

a) оповещения об изменениях зависящих объектов

b) взаимодействия множества объектов

c) уровней ответственности в системе

d) независимости классов отправителя и получателя

75. Ниже приведенный код представляет собой ...

```
class C
```

```
{
```

```
public:
```

```
virtual void Execute() = 0;
```

```
virtual void unExecute() = 0;
```

```
};
```

```
class B : public C
```

```
{
```

```
public:
```

```
void Execute(){/**/}
```

```
void unExecute(){/**/}
```

```
};
```

```
class Receiver
```

```
{
```

```

C * c;
public:
void Redo()
{
c = new B ();
c->Execute();
}

void Undo()
{
c->unExecute();
delete command;
}

};

```

- a) Стратегия
- b) Декоратор
- c) Команда
- d) Наблюдатель

76. В библиотеки STL есть возможность обойти последовательно все элементы некоторого контейнера. Какой паттерн проектирования был использован для достижения этой цели?

- a) Visitor
- b) Observer
- c) Iterator
- d) Walker
- e) Collector

77. Результатом применения какого паттерна является класс iterator в библиотеке STL ?

- a) Factory Method
- b) Abstract Method
- c) Singleton
- d) Template Method
- e) Iterator

78. Итератор относится к шаблонам.

- a) порождающим
- b) структурным
- c) поведенческим
- d) основным

79. Итератор представляет собой объект, позволяющий ...

- a) получить последовательный доступ к элементам
- b) предотвращать нулевые указатели
- c) сохранить внутреннее состояние объекта
- d) взаимодействовать множество объектов

80. Главное преимущество шаблона итератор является ...

- a) раскрытие внутреннего представления объекта
- b) не раскрытие внутреннего представления объекта

- c) последовательный доступ к элементам
- d) предотвращение нулевых указателей

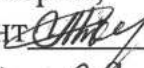
Ответы

ДЕ-1	
1	a, b
2	c
3	a
4	d
5	c
6	c
7	d
8	a, b
9	a
10	b
ДЕ-2	
11	a, c, d
12	c
13	a
14	a
15	c
16	d
17	b
18	d
19	c
20	d
21	c
22	b
23	a
24	a
25	b
26	b, c
27	c
ДЕ-3	
28	a, c
29	a, b
30	a, b, d
31	d
32	d
33	a
34	c
35	b
36	a
37	d
38	d
39	c

40	c
41	d
42	a, d
43	c
44	b
45	c
46	b, d
47	c
48	a
49	b
50	a, b
51	c
52	a
53	b
54	b
55	d
56	d
57	b
58	c
59	b
60	d
61	a
62	b
63	c
64	c
65	b
66	a
67	b
68	c
69	D
70	A
71	D
72	c
73	a
74	d
75	c
76	c
77	e
78	c
79	a
80	b

«УТВЕРЖДАЮ»

зав. кафедрой информатики и программной инженерии,

доцент  Л.А. Тягульская

«19» 09 2024 г.

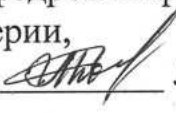
Вопросы к экзамену
по дисциплине «Проектирование программного обеспечения»
для студентов II курса
направления подготовки «Программная инженерия»
профиль подготовки «Разработка программно-информационных систем»,
IV семестр

1. Программные и пользовательские исключения. Оператор throw.
2. Обработка исключений. Операторы try..catch.
3. Библиотека iostream. Поточный ввод/вывод данных.
4. Строки C++. Стандартный класс string.
5. Понятие потока.
6. Разработка многопоточных приложений.
7. Стандартная библиотека шаблонов STL.
8. Преимущества и недостатки STL.
9. Контейнеры стандартной библиотеки STL.
10. Распределители памяти стандартной библиотеки STL.
11. Итераторы стандартной библиотеки STL.
12. Алгоритмы стандартной библиотеки STL.
13. Контейнер вектор.
14. Контейнер список.
15. Паттерны проектирования.
16. Общие принципы паттернов проектирования
17. Описание паттернов проектирования
18. Шаблон делегирования
19. Шаблон «Интерфейс».
20. Шаблон «Абстрактная фабрика».
21. Шаблон «Строитель».
22. Шаблон «Адаптер».
23. Шаблон «Мост»
24. Шаблон «Цепочка обязанностей».
25. Шаблон «Команда»
26. Шаблон «Итератор»

Экзаменатор, преподаватель  О.М. Нагаевский

«УТВЕРЖДАЮ»

зав. кафедрой информатики и программной инженерии,

доцент  Л.А. Тягульская

«__» _____ 2024 г.

**Практические задания к экзамену
по дисциплине «Проектирование программного обеспечения»
для студентов II курса
направления «Программная инженерия»
профиля подготовки
«Разработка программно-информационных систем»,
IV семестр**

Практическое задание № 1

Реализовать шаблон «Строитель» на примере фабрики по созданию мобильных телефонов. Для простоты считать, что телефон может состоять из корпуса, CDMA модуля, GSM модуля, антенны и экрана. Реализовать возможность создания CDMA и GSM телефонов.

Практическое задание № 2

Используя шаблон «Адаптер» реализовать класс, который считывает данные о размере диагонали телевизора в сантиметрах, а выдает данные в дюймах.

Практическое задание № 3

Дан текстовый файл f. Вывести на экран слово, которое повторяется наибольшее количество раз.

Практическое задание № 4

Заполнить контейнер stack числами из файла и удалить все элементы кратные 3-м. Вывести содержимое стека на экран.

Практическое задание № 5

Используя шаблон делегирования реализовать программу с выводом заданной строки в Edit и Memo.

Практическое задание № 6

Реализовать шаблон «Цепочка обязанностей» на примере обработки бинарного файла в зависимости от типа файла. Для простоты в системе существуют три типа документов исполняемый, текстовый и мультимедийный (exe, txt, mp4).

Практическое задание № 7

Реализуйте класс для восьмеричных чисел в текстовом формате. Обработайте ошибки динамического выделения памяти. Обработайте ошибки задания числа (допустимый формат «только цифры 0-7»).

Практическое задание № 8

Реализовать шаблон «Строитель» на примере фабрики по созданию телевизоров. Для простоты считать, что телевизор может состоять из корпуса, матрицы, подсветки, DVI интерфейс, VGA интерфейс. Реализовать возможность создания телевизоров с DVI и VGA интерфейсами.

Практическое задание № 9

Реализовать шаблон «Абстрактная фабрика» на примере фабрики по созданию костюмов. Для простоты считать, что костюм состоит из пиджака и брюк и что существуют две разные фирмы по созданию брюк

Практическое задание № 10

Реализовать шаблон «Фасад» на примере создания html странички. Предоставить пользователю простой метод создания html странички, реализующий создание из составных частей.

Практическое задание № 11

Создать программу, высчитывающую во втором потоке наибольшее простое число от 1 до 10000 и выводящее результат в консоль

Практическое задание № 12

Используя паттерн «Итератор» реализовать доступ файлам в папке.

Практическое задание № 13

Реализовать шаблон «Фасад» на примере создания компьютера. Предоставить пользователю простой метод создания компьютера, реализующий создание из составных частей.

Практическое задание № 14

Реализовать шаблон «Абстрактная фабрика» на примере фабрики по созданию автомобилей. Для простоты считать, что автомобиль состоит из двигателя, кузова и салона. Предположим, что существуют две разные фирмы по созданию автомобилей Ford и BMW

Практическое задание № 15

Дан текстовый файл со словами разделенными пробелами. Считать файл в вектор и отсортировать слова по возрастанию. Результат вывести на экран.

Практическое задание № 16

Реализовать шаблон «Мост» на примере автоматической синхронизации данных в поле ввода (Метод) с другим полем и с заданным файлом.

Практическое задание № 17

Создать интерфейс класса для воина, из какой либо игры.