

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 «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ»

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

« 18 » 09 2024г.

Рыбница 2024г.

Паспорт фонда оценочных средств по учебной дисциплине

1. В результате изучения дисциплины «Объектно-ориентированное программирование» 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-коммуникационных технологий и с учетом	ИД ОПК-3.1. Знает принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований

	основных требований информационной безопасности	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.	Разработка приложений под ОС Microsoft Windows	УК-1, УК-3, ОПК-1, ОПК-2, ОПК-3, ОПК-6, ОПК-7, ПК-5, ПК-6	Лабораторные работы. Тестирование.
3	Промежуточная аттестация	УК-1, УК-3, ОПК-1, ОПК-2, ОПК-3, ОПК-6, ОПК-7, ПК-5, ПК-6	Экзамен.

«УТВЕРЖДАЮ»

зав. кафедрой ИиПИ,

доцент  Л. А. Тягульская

« 19 » 09 2024 г.

Итоговый тест по дисциплине
«Объектно-ориентированное программирование»
для студентов II курса
направления «Программная инженерия» профиля подготовки
«Разработка программно-информационных систем»

ДЕ1

1. Программа – это

- a) протокол взаимодействия компонентов компьютерной сети;
- b) набор команд операционной системы компьютера;
- c) алгоритм, записанный на языке программирования;
- d) законченное минимальное смысловое выражение на языке программирования.

2. Процесс написания программы никогда не включает

- a) процесс отладки;
- b) редактирование текста программы;
- c) изменение физических параметров компьютера;
- d) запись операторов соответствующей языку форме.

3. Деятельность, направленная на обнаружение и исправление ошибок в программной системе называется

- a) тестированием;
- b) отладкой;
- c) демонстрацией;
- d) рефакторингом.

4. На этапе отладки программы

- a) строится математическая модель решаемой задачи;
- b) проверяется корректность работы программы;
- c) выполняется анализ физических характеристик;
- d) определяется состав входных данных.

5. Какой подход в программировании называется структурным?

- a) подход, ориентированный на непосредственно выполняемые компьютером операции;
- b) подход, неориентированный на непосредственно выполняемые компьютером операции;
- c) подход, ориентированный на то, что отдельные группы операторов могут объединяться во вспомогательные алгоритмы;
- d) подход, ориентированный на то, что логическая структура программы может быть выражена комбинацией 3-х базовых структур — следования, ветвления, цикла.

6. Основой метода структурного программирования являются ...

- a) принцип модульности разработки сложных программ;
- b) использование композиции трех базовых элементов – линейной, ветвления и циклической ;
- c) использование композиции двух базовых элементов – ветвления и циклической;

d) использование большого количества подпрограмм.

7. Правила композиции, используемые при структурном подходе к составлению алгоритмов:

- a) альтернативный выбор;
- b) цикл;
- c) подпрограммы;
- d) метки.

8. Модульная структура программы отражает одну из особенностей _____ программирования

- a) эвристического;
- b) структурного;
- c) логического;
- d) динамического.

9. Для чего нужны классы?

- a) для определения новых типов в программе;
- b) для упрощения работы с функциями;
- c) для соединения данных и операций над ними;
- d) для упрощения работы с указателями.

10. Методы класса определяют:

- a) какие операции можно выполнять с объектами данного класса;
- b) какие типы значений могут принимать данные-члены класса;
- c) каким образом реализуется защита данных-членов класса.

11. Атрибуты (данные-члены) класса могут быть:

- a) только целыми числами;
- b) любыми встроенными типами;
- c) любого определенного в программе типа.

12. Какая из записей соответствует обращению к атрибуту arg класса A в определении метода этого же класса?

- a) `this -> arg;`
- b) `arg;`
- c) `A -> arg;`
- d) `A -> this -> arg.`

13. Если имеется код

`class A {public: int a; }; A *obj;` то как обратиться к переменной a?

- a) `obj.a;`
- b) `(*obj) -> a;`
- c) `obj -> a;`
- d) `obj :: a.`

14. Если имеется код

`class A {public: int a, b, c;}; A *obj;` то как обратиться к переменной b?

- a) `(*obj) -> b;`
- b) `A :: b;`
- c) `(*obj).b;`
- d) `obj -> a.b.`

15. Каков будет результат выполнения следующего кода:

```

class A {
public:
int inc (int x) {return x++;}
int inc (short x) (return x+2;}
};
A obj; int y=5;
cout << obj.inc(y); ?

```

- a) 5;
- b) 6;
- c) 7;
- d) ошибка при компиляции.

16. Каков будет результат выполнения следующего кода:

```

class A {
public:
int y;
int inc (int x) {return y++;}
int inc (short x) {return x+y;}
};
A obj; int y=5; obj.y= 6;
cout << obj.inc(y); ?

```

- a) 5;
- b) 6;
- c) 11;
- d) 7;
- e) ошибка при компиляции.

17. Какими по умолчанию объявляются элементы структуры?

- a) private;
- b) public;
- c) protected;
- d) по умолчанию не объявляются.

18. Какие из операторов не могут быть перегружены?

- a) new;
- b) .;
- c) *;
- d) [];
- e) ().

19. Отметьте все утверждения, которые считаете верными:

- a) нельзя с помощью шаблона создать функцию с таким же именем, как у явно определенной функции;
- b) цель введения шаблонов – создание функций, которые могут обрабатывать однотипные данные;
- c) при использовании шаблонов функций возможна перегрузка как с помощью шаблонов, так и функций;
- d) в качестве описания шаблона функции используется прототип шаблона;
- e) Template<список_параметров_шаблона>

20. Можно ли задать шаблон со значением параметра по умолчанию?

- a) да;
- b) нет.

21. Каков правильный заголовок шаблона?

- a) `template<class t1,class t2>;`
- b) `template <class t1,t2>;`
- c) `template <class t, class t>.`

22. Сколько параметров может быть у шаблона при определении шаблона функции?

- a) 1;
- b) столько, сколько аргументов у функции;
- c) столько, сколько типов используется для параметризации.

23. Отметьте правильный вариант описания шаблона семейства функций:

- a) `template(class T)`
`void func(T* p1,T* p2){...}`
- b) `template <class T>;`
`void func(T* p1,T* p2){...}`
- c) `template<class T>`
`void func(T* p1,T* p2){...}`
- d) `template<class T>`
`void func(T* p1,T* p2){...}`

24. Можно ли использовать класс-шаблон в качестве базового класса?

- e) да;
- f) нет.

25. Какая из записей является правильной записью абстрактного класса?

- a) `abstract class A{virtual f()=0;};`
- b) `class A{virtual f()=0;};`
- c) `class A{virtual f();}.`

26. Абстрактный класс – это класс, в котором:

- a) есть хотя бы один виртуальный метод;
- b) есть виртуальный конструктор;
- c) есть виртуальный деструктор;
- d) есть чисто виртуальный метод.

27. Основная проблема множественного наследования состоит в:

- a) замедлении выполнения программ;
- b) необходимости явно указывать, к какому базовому классу принадлежит метод;
- c) возможности существования двух экземпляров объекта базового класса;
- d) неэкономном расходовании памяти.

28. Если записано

```
class A {public: void f(){cout<<1;}};  
class B:public A {public: void f(){cout<<2;}};
```

то что будет напечатано, если

`B b; b.f();` ?

- a) 2;
- b) 21;
- c) 12;
- d) 1.

29. Если записано

```
class A{public:virtual void f(){cout<<1;}};  
class B:public A{public:virtual void f(){cout<<2;}};
```

то что будет напечатано, если

B b; b.f(); ?

- a) 2;
- b) 21;
- c) 12;
- d) 1;
- e) ошибка.

30. Если имеется код `char a[8]; cin>>a;` и вводится текст "Hello World", то что будет в массиве a?

- a) "Hello W";
- b) "Hello Wo";
- c) "Hello";
- d) "Hello World";
- e) "lo World".

31. Что будет выведено в результате

`double x=12.4;`

`cout<<setw(5)<<x<<setw(3)<<setfill('*')<<" "<<endl; ?`

- a) 12.40***;
- b) 12.4***;
- c) 12.4 ***;
- d) 12.40;
- e) 124e2***.

32. Если имеется код `int x; cin>>x;` и вводится "1.2", что будет в переменной x?

- a) 1;
- b) 2;
- c) 1.2;
- d) другое;
- e) произойдет ошибка.

33. Какой из классов используется для вывода строк на экран?

- a) `stringstream`;
- b) `ostream`;
- c) `ofstream`;
- d) `istream`;
- e) `ifstream`.

34. Каким будет результат работы программы:

```
#include <iostream.h>
```

```
void main (){
```

```
char A[]="ABC";
```

```
char *U=&A[2];
```

```
cout<<"\n"<<*U--<<*U--<<*U<<endl; } ?
```

- a) BAA;
- b) CBA.

35. Если имеется код `double x; cin>>x;` и вводится "12-3", то что будет в переменной x?

- a) 9.0;
- b) 12.0;
- c) другое;
- d) произойдет ошибка.

36. Для того, чтобы выполнить чтение из файла с произвольной позиции, надо использовать объект класса:

- a) stringstream;
- b) ostream;
- c) ofstream;
- d) istream;
- e) ifstream.

37. Об ошибке в конструкторе класса может сигнализировать:

- a) возвращаемое значение;
- b) исключительная ситуация;
- c) вызов деструктора сразу в конструкторе.

38. Оператор throw без аргументов

- a) повторно вызывает обрабатываемую исключительную ситуацию;
- b) вызывает исключительную ситуацию типа Exception.

ДЕ-2.

39. Основным элементов графического интерфейса операционной системы Windows НЕ является _____.

- a) рабочий стол
- b) командное окно
- c) значок
- d) панель

40. Основным элементом графического интерфейса, который обеспечивает эффективный доступ пользователя ко всем ресурсам компьютера, к наиболее часто используемым программам, является _____.

- a. окно
- b. рабочий стол
- c. значок
- d. панель

41. Элементом графического интерфейса пользователя, который служит для объединения группы логически взаимосвязанных значков, является _____.

- a) меню
- b) рабочий стол
- c) кнопка
- d) панель

42. Элементом графического интерфейса пользователя, представляющий собой список команд, действия, является _____.

- a) меню
- b) рабочий стол
- c) значок
- d) ярлык

43. Элементом графического интерфейса пользователя, предназначенного для группировки файлов по какому-либо признаку, является _____.
- a) меню
 - b) папка
 - c) значок
 - d) ярлык
44. Элементом графического интерфейса пользователя, принимающим управляющее воздействие пользователя и запускающим некоторую последовательность действия, является _____.
- a) меню
 - b) папка
 - c) значок
 - d) кнопка
45. Средством взаимодействия между окнами являются _____
- a) сообщения
 - b) события
 - c) письма
 - d) файлы
46. Восстановите правильную последовательность обработки сообщения Windows
- a) TranslateMessage
 - b) GetMessage
 - c) DispatchMessage
47. Для возвращения очередного сообщения Windows используются функции _____ (выберите один или несколько вариантов ответа)
- a) TranslateMessage
 - b) GetMessage
 - c) DispatchMessage
 - d) PeekMessage
48. Для обработки сообщения Windows используется функция _____
- a) TranslateMessage
 - b) GetMessage
 - c) DispatchMessage
 - d) PeekMessage
49. Для отправки сообщения окну используется функция _____.
- a) TranslateMessage
 - b) GetMessage
 - c) DispatchMessage
 - d) SendMessage
50. При закрытие окна генерируется сообщение _____
- a) WM_CREATE
 - b) WM_CLOSE
 - c) WM_ACTIVATE
 - d) WM_COMMAND
51. C++ Builder является _____
- a) системой разработки приложений

- b) текстовым редактором
- c) компилятором
- d) языком программирования

52. Палитра компонентов используется для _____

- a) изменения свойств компонентов
- b) изменения реакций компонентов на события
- c) просмотра объектов, добавленных на форму
- d) добавления компонентов на форму

53. Окно инспектора объектов используется для _____ (выберите один или несколько вариантов ответа)

- a) изменения свойств компонентов
- b) изменения реакций компонентов на события
- c) просмотра объектов, добавленных на форму
- d) добавления компонентов на форму

54. Окно просмотра объектов используется для _____

- a) изменения свойств компонентов
- b) изменения реакций компонентов на события
- c) просмотра объектов, добавленных на форму
- d) добавления компонентов на форму

55. Основой графического приложения в C++ Builder 6 является _____

- a) форма
- b) кнопка
- c) окно
- d) панель

56. Для создания консольного приложения в C++ Builder 6 необходимо выполнить команду _____

- a) File->New->Other - Console Wizard
- b) File->Application
- c) File->CLX Application
- d) File->Form

57. Для создания графического приложения Windows в C++ Builder 6 необходимо выполнить команду _____

- a) File->New->Other - Console Wizard
- b) File->Application
- c) File->CLX Application
- d) File->Form

58. Для создания динамически подключаемой библиотеки (dll) в C++ Builder 6 необходимо выполнить команду _____

- a) File->New->Other - Dll Wizard
- b) File->Application
- c) File->New->Other - Console Wizard
- d) File->New->Other - Library

59. По умолчанию, исходный код графического приложения в C++ Builder 6.0 содержится в файле с расширением _____

- a) .bak
- b) .bpr
- c) .bpf
- d) .cpp

60. Файл формы графического приложения в C++ Builder 6.0 имеет расширение _____

- a) .dfm
- b) .bpr
- c) .bpf
- d) .cpp

61. Файл проекта графического приложения в C++ Builder 6.0 имеет расширение _____

- a) .dfm
- b) .bpr
- c) .bpf
- d) .cpp

62. Файл графического приложения в C++ Builder 6.0, содержащий описание классов формы, имеет расширение _____

- a) .dfm
- b) .bpr
- c) .h
- d) .cpp

63. Базовым для всех VCL классов является _____

- a) TComponent
- b) TWinControl
- c) TObject
- d) TCustomControl

64. Базовым для всех VCL компонент является класс _____

- a) TComponent
- b) TWinControl
- c) TObject
- d) TCustomControl

65. Оконные элементы, компоненты стандартного управления произведены непосредственно от класса _____

- a) TComponent
- b) TWinControl
- c) TObject
- d) TCustomControl

66. Невидимые компоненты произведены от класса _____

- a) TComponent
- b) TWinControl
- c) TObject
- d) TCustomControl

67. Графические компоненты, не ассоциированные с оконными элементами управления, произведены от класса _____

- a) TGraphicControl

- b) TWinControl
- c) TObject
- d) TCustomControl

68. Примером невидимого компонента является _____

- a) TEdit
- b) TForm
- c) TOpenDialog
- d) TPanel

69. Для выравнивания графического компонента внутри родителя используется свойство _____

- a) Alignment
- b) Align
- c) Anchors
- d) AutoScroll

70. Для изменения имени компонента необходимо использовать свойство _____.

- a) Caption
- b) Text
- c) Name
- d) FileName

71. Для изменения текста в компоненте TEdit необходимо использовать свойство _____

- a) Caption
- b) Text
- c) Name
- d) FileName

72. Для изменения текста заголовка TForm необходимо использовать свойство _____

- a) Caption
- b) Text
- c) Name
- d) Title

73. Для добавления в программу выпадающего списка текстовых строк используется компонент _____

- a) TComboBox
- b) TListBox
- c) TMemo
- d) TPopupMenu

74. Компонент, представляющий собой кнопку с рисунком, является _____

- a) TButton
- b) TBitBtn
- c) TSpeedButton
- d) TRadioButton

75. Компонент, который НЕ является стандартным, это диалог _____

- a) открытия файла
- b) поиска файла
- c) открытия картинки

d) выбора цвета

76. Для выполнения стандартного диалога необходимо вызвать метод _____

- a) Go
- b) Use
- c) Run
- d) Execute

77. Для получения имени файла из диалога открытия (OpenDialog) необходимо использовать свойство _____

- a) FileName
- b) File
- c) Name
- d) Filter

78. Для отображения только исполняемых файлов (*.exe) необходимо редактировать свойство _____

- a) FileName
- b) DefaultExt
- c) Name
- d) Filter

79. Основным свойством диалога выбора шрифта является _____

- a) MaxFontSize
- b) MinFontSize
- c) Font
- d) Name

80. Для получения цвета из диалога выбора цвета используется поле _____

- a) CustomColors
- b) Name
- c) Execute
- d) Color

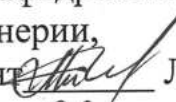
Отвѣты

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

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

«УТВЕРЖДАЮ»

зав. кафедрой информатики и программной инженерии,

доцент  Л.А. Тягульская

« 19 » 09 2024 г.

Вопросы к экзамену
по дисциплине «Объектно-ориентированное программирование»
для студентов II курса
направления подготовки «Программная инженерия»
профиль подготовки «Разработка программно-информационных систем»,
III семестр

1. Понятие Класса и Объекта
2. Наследование и типы наследования
3. Инкапсуляция и уровни доступа
4. Полиморфизм
5. Перегрузка функции
6. Перегрузка операндов
7. Функции с параметрами по умолчанию
8. Функции с переменным числом параметров
9. Дружественность. Дружественные методы, функции и классы. Наследование дружбы
10. Абстрактные классы и виртуальные функции.
11. Виртуальные деструкторы и порядок вызова конструкторов и деструкторов
12. Конструкторы и деструкторы.
13. Конструкторы копирования и конструкторы по умолчанию.
14. Статические члены класса.
15. Константные члены класса и объекты
16. Шаблоны функции и классов.
17. Подключение динамической библиотеки к проекту
18. Поток выполнения данных
19. Динамически подключаемые библиотеки DLL
20. Понятие потока в C++ и его использование

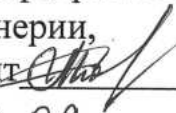
Экзаменатор, преподаватель



О.М. Нагаевский

«УТВЕРЖДАЮ»

зав. кафедрой информатики и программной инженерии,

доцент  Л.А. Тягульская

« 19 » 09 2024 г.

**Практические задания к экзамену
по дисциплине «Объектно-ориентированное программирование»
для студентов II курса
направления «Программная инженерия»,
III семестр**

Практическое задание № 1

Создать перегруженную функцию выводящую информацию в консоль, которая удовлетворяет следующим условиям:

при передаче одного параметра типа string функция просто выводит данный текст;

при передаче двух параметров типа string первый параметр считается заголовком и выводится в отдельной строке, затем добавляется строка с вторым параметром в качестве текста;

при передаче одного параметра типа int в консоль выводится данное число в десятичном, восьмеричном и десятичном видах.

Практическое задание № 2

Создать шаблон класса для массива элементов, создать конструктор копирования, конструктор по умолчанию и перегрузить оператор []

Практическое задание № 3

Создать класс для работы с прямоугольником и перегрузить для него операторы +, -, *

Практическое задание № 4

Создать шаблон класса, реализующий дек. Дек – это структура данных, которая позволяет удалять и добавлять элементы, как в начало, так и в конец.

Практическое задание № 5

Создать абстрактный класс «Геометрическая фигура». Создать три класса наследника: «Круг», «Линия» и «Прямоугольник». Реализовать в наследниках абстрактные функции класса «Геометрическая фигура». Продемонстрировать полиморфизм.

Практическое задание № 6

Создать класс «Прямоугольник», сделав все поля класса приватными. Создать дружественный класс, который позволит менять приватные поля класса «Прямоугольник».

Практическое задание № 7

Создать класс «Строка» с конструкторами копирования, принимающими в себя целые и дробные числа, строки и символы.

Практическое задание № 8

Создать класс подсчета факториала числа, причем расчет производить в разных объектах данного класса и для хранения общего результата использовать статическое поле.

Практическое задание № 9

Создать класс «Круг», в котором реализовать метод расчета площади круга. Число π реализовать в качестве константного поля. Реализовать константный и не константный метод вывода площади. Продемонстрировать работу класса для константных и не константных объектов.

Практическое задание № 10

Создать класс «Линия» с конструктором по умолчанию, принимающим в себя координаты начальной точки и длину. Реализовать методы изменения длины, перемещения и поворота линии. Создать массив объектов типа «Линия». В качестве примера передвинуть на 10 единиц, изменить размер на 2 единицы и повернуть каждую линию.

Практическое задание № 11

Создать классы 2-3 класса машин (легковой, грузовой, пассажирский) с использованием наследование и интерфейсов. Реализовать конструктор копирования и конструктор по умолчанию.

Практическое задание № 12

Создать иерархию классов для различных типов самолетов (штурмовик, бомбардировщик, разведчик) с использованием наследование и интерфейсов. Реализовать конструктор копирования и конструктор по умолчанию. Продемонстрировать полиморфизм.

Практическое задание № 13

Создать иерархию классов для различных типов визуализации данных (график, гистограмма, круговая диаграмма) с использованием наследование и интерфейсов. Реализовать конструктор копирования и конструктор по умолчанию. Продемонстрировать полиморфизм.

Практическое задание № 14

Создать иерархию классов для различных типов космических объектов (планета, звезда, галактика) с использованием наследование и интерфейсов. Реализовать конструктор копирования и конструктор по умолчанию. Продемонстрировать полиморфизм.

Практическое задание № 15

Создать иерархию классов для различных типов кораблей (буксир, круизный лайнер, авианосец) с использованием наследование и интерфейсов. Реализовать конструктор копирования и конструктор по умолчанию. Продемонстрировать полиморфизм.

Практическое задание № 16

Создать классы 2-3 класса машин (легковой, грузовой, пассажирский) с использованием наследование и интерфейсов. Реализовать конструктор копирования и конструктор по умолчанию.

Практическое задание № 17

Создать иерархию классов для различных типов самолетов (штурмовик, бомбардировщик, разведчик) с использованием наследование и интерфейсов. Реализовать конструктор копирования и конструктор по умолчанию. Продемонстрировать полиморфизм.

Практическое задание № 18

Создать иерархию классов для животных 3-4 класса. Реализовать конструктор копирования и конструктор по умолчанию. Продемонстрировать полиморфизм.

Практическое задание № 19

Создать шаблон класса для трехмерных векторов, реализовать конструкторы копирования и по умолчанию, перегрузить операторы $+$, $-$, $*$ и $/$

Практическое задание № 20

Создать классы воинов(лучник, стрелок, пехотинец) с использованием наследование и интерфейсов. Реализовать конструктор копирования и конструктор по умолчанию.

Экзаменатор, преподаватель



О.М. Нагаевский