Перечень теоретических и практических заданий к экзамену по ОП.04 Основы алгоритмизации и программирования (2 курс, 4 семестр 2022-2023 уч. г.)

Форма контроля: Практическая работа (Опрос)
Описательная часть: ответьте на 15 вопросов теста и выполните одно практическое задание

Перечень теоретических заданий:
Задание №1

Вопрос 1 Для решения любой задачи с помощью компьютера необходимо выполнить следующие этапы:

Установите правильную последовательность этапов.

1	Принятие решений
2	Программирование
3	Математическое моделирование
4	Алгоритмизация задач
5	Постановка задачи
6	Анализ результатов

Вопрос 2 *Соотнесите свойства алгоритма с их описанием:*

Результативность	- алгоритм должен приводить к решению задачи обязательно за конечное время
Конечность	- неоднозначность толкования алгоритма недопустима
Эффективность	- алгоритм должен обеспечить выдачу результата решения задачи на печать, на экран монитора или в файл
Массовость	- правильный результат по алгоритму получен для одних исходных данных,то правильный результат по этому же алгоритму должен быть получен и для других исходных данных, допустимых в данной задаче

Определенность	- позволяет решить задачу за приемлемое для разработчика время

Вопрос 3 Выберите тип алгоритма, описанного ниже:

Набор команд (указаний), выполняемых последовательно друг за другом

- Линейный
- Разветвляющийся
- Циклический

Вопрос 4 Выберите тип алгоритма, описанного ниже:

Алгоритм, содержащий хотя бы одну проверку условия, в результате которой обеспечивается переход на один из возможных вариантов решения

- Линейный
- Разветвляющийся
- Циклический

Вопрос 5 Выберите тип алгоритма, описанного ниже:

Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными

- Линейный
- Разветвляющийся
- Циклический

Оценка	Показатели оценки
3	Даны правильные ответы на 3 вопроса
4	Даны правильные ответы на 4 вопроса
5	Даны правильные ответы на 5 вопросов

Задание №2

Вопрос 1 Для решения любой задачи с помощью компьютера необходимо выполнить следующие этапы:

Установите правильную последовательность этапов.

1	Принятие решений
2	Программирование
3	Математическое моделирование
4	Алгоритмизация задач
5	Постановка задачи
6	Анализ результатов

Вопрос 2 *Соотнесите свойства алгоритма с их описанием:*

Результативность	- алгоритм должен приводить к решению задачи обязательно за конечное время
Конечность	- неоднозначность толкования алгоритма недопустима
Эффективность	- алгоритм должен обеспечить выдачу результата решения задачи на печать, на экран монитора или в файл
Массовость	- правильный результат по алгоритму получен для одних исходных данных,то правильный результат по этому же алгоритму должен быть получен и для других исходных данных, допустимых в данной задаче
Определенность	- позволяет решить задачу за приемлемое для разработчика время

Вопрос 3 Выберите тип алгоритма, описанного ниже:

Набор команд (указаний), выполняемых последовательно друг за другом

- Линейный
- Разветвляющийся
- Циклический

Вопрос 4 Выберите тип алгоритма, описанного ниже:

Алгоритм, содержащий хотя бы одну проверку условия, в результате которой обеспечивается переход на один из возможных вариантов решения

- Линейный
- Разветвляющийся
- Циклический

Вопрос 5 Выберите тип алгоритма, описанного ниже:

Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными

- Линейный
- Разветвляющийся
- Циклический

Оценка	Показатели оценки

Задание №3

Вопрос 1 Установите правильную хронологию создания языков программирования:

1	C#
2	Бейсик (Basic)
3	Паскаль (Pascal)
4	C++

Вопрос 2 Соотнесите годы создания языков программирования

1963	C#
1971	Бейсик (Basic)
1984	Паскаль (Pascal)
2000	C++

Вопрос 3 Выберите процедурный язык программирования:

- C++
- Basic
- Java

Вопрос 4 Какие языки программирования предназначены для решения задач искусственного интеллекта:

- Commonlisp
- Planner
- Occam
- C++
- Java

Вопрос 5 Какие языки программирования предназначены для разработки программ-оболочек, разработки систем:

- Commonlisp
- Planner
- Occam
- C++
- Java

Оценка 345

Оценка 345

Оценка 345

Оценка 345

Оценка 345

Оценка 345

Оценка	Показатели оценки
3	Даны правильные ответы на 3 вопроса
4	Даны правильные ответы на 4 вопроса
5	Даны правильные ответы на 5 вопросов

Задание №4

Вопрос 1 Установите правильную хронологию создания языков программирования:

1	C#
2	Бейсик (Basic)

3	Паскаль (Pascal)
4	C++

Вопрос 2 Соотнесите годы создания языков программирования

1963	C#
1971	Бейсик (Basic)
1984	Паскаль (Pascal)
2000	C++

Вопрос 3 Выберите процедурный язык программирования:

- C++
- Basic
- Java

Вопрос 4 Какие языки программирования предназначены для решения задач искусственного интеллекта:

- Commonlisp
- Planner
- Occam
- C++
- Java

Вопрос 5 Какие языки программирования предназначены для разработки программ-оболочек, разработки систем:

- Commonlisp
- Planner
- Occam
- C++
- Java

Оценка 345

Оценка 345

Оценка 345

Оценка 345

Оценка 345

Оценка 345

Оценка	Показатели оценки

Задание №5

Вопрос 1 Выберите правильный вариант использования условного оператора if для нахождения MAX{C*D, E+F}:

- if (C*D>E+F) Max=C*D; else Max:=E+F;
- X=C*D; Y=E+F; if (X>Y) Max=X; else MAX=Y;
- оба варианта правильные

Вопрос 2 Выберите пример правильного идентификатора в языке C++

- Fr_5
- 10Sd
- scanf

Вопрос 3 Какие значения примут переменные S, N и P после выполнения следующего фрагмента программы,

если в переменную Y поочередно ввести следующие значения: -2; 0; -5; 6; 0?

...

S=0; N=0; P=1;

for (i=1; i<=5; i++)

{

scanf ("%d", &Y);

if (Y>0)

S=S+Y;

else if (Y=0)

N=N+1;

else P=P*Y;

}

...

- S=-7; N=1; P=0;
- S=6; N=2; P=10;
- S=-1; N=4; P=10;

Вопрос 4 Что делает следующий фрагмент программы с массивом A, содержащим N элементов?

...

for (i=0; i<N/2; i++)

{ M=A[i];

A[i]=A[N-1-i];

A[N-1-i]=M;

}

...

- присваивает каждому элементу массива значение следующего элемента, а последнему - значение 1-го элемента
- записывает элементы массива в обратном порядке
- фрагмент содержит ошибки и работать не будет

Вопрос 5. Укажите правильный вариант определения класса в программе:

- class Test
  {
  public:
  int a;
  };
- class Test
  {

private:
float x;
}

- оба варианта правильные.

Оценка	Показатели оценки
3	Даны правильные ответы на 3 вопроса
4	Даны правильные ответы на 4 вопроса
5	Даны правильные ответы на 5 вопросов

Задание №6

Вопрос 1. В чем разница между формальными и фактическими параметрами?

- никакой разницы нет;
- формальные параметры используются при описании подпрограммы, а фактические — при вызове подпрограммы;
- фактические параметры используются при описании подпрограммы, а формальные — при вызове подпрограммы.

Вопрос 2. В чем разница между глобальными и локальными переменными?

- разницы нет;
- глобальные переменные не могут использоваться в подпрограммах, для этого служат локальные переменные;
- глобальные переменные могут использоваться во всех подпрограммах и в функции main (), а локальные переменные только в своей подпрограмме.

Вопрос 3. Выберите определение функции, не содержащее ошибок:

- int Area(int A, int B)
  { float S;
  Area=A*B;}

Оценка 345

Оценка 345

- int Area(int A, int B)
  { int S;
  S=A*B; return S;}

- int Area(int A, int B)
  { int S;
  S=A*B;
  return Area;

Вопрос 4. Что будет выведено на экран в результате работы следующей программы?

```
int A;

void Prim(int A)

{A=5;

printf(" %d",A);

}

int main()

(int A=10;

Prim(A);

printf ( " %d", A) ; return 0;

}
```

- 10 10;
- 5 10;
- 5 5.

Вопрос 5. Должны ли имена параметров, указанных в прототипе, определении и вызове функции, соответствовать друг другу?

- да;
- нет;
- должны соответствовать в прототипе и определении.

Оценка	Показатели оценки
3	Даны правильные ответы на 3 вопроса
4	Даны правильные ответы на 4 вопроса
5	Даны правильные ответы на 5 вопросов

Задание №7

Вопрос 1. Принцип инкапсуляции обеспечивает:

- объединение данных и методов работы с ними в классе;
- доступ к членам класса;
- сокрытие данных внутри класса.

Вопрос 2. Укажите правильный вариант инициализации целочисленных констант A и B в классе:

- klass::klass(int a, int b) : A(a), B(b)
  {
  • • •
  }

Оценка 345

Оценка 345

- klass::klass(int a, int b) : A= a, B = b
  {
  • • •
  }
- klass::klass(int a, int b) : A(a) B(b)
  {
  • • •
  }

Вопрос 3. Наследование — это:

- возможность использования базовых библиотек языка C++ в своих программах;
- условия, описывающие последовательность вызова конструкторов для объектов классов, используемых в программе;
- механизм создания производных классов, на базе уже имеющихся.

Вопрос 4. Укажите верную последовательность выполнения деструкторов:

- сначала выполняются операторы деструктора базового класса, затем операторы деструктора порожденного класса;
- сначала выполняются операторы деструктора порожденного класса, затем операторы деструктора базового класса;
- операторы деструктора порожденного класса выполняются одновременно с операторами деструктора базового класса.

Вопрос 5. Полиморфизм — это:

- возможность программного кода работать с разными объектами одинаковым образом;
- возможность изменения программного кода в зависимости от решаемых задач;
- возможность доработки программного кода в случае необходимости.

Оценка	Показатели оценки
3	Даны правильные ответы на 3 вопроса
4	Даны правильные ответы на 4 вопроса
5	Даны правильные ответы на 5 вопросов

Перечень практических заданий:
Задание №1

Разработайте алгоритм для одной из задачи:

Задача 1. Определить, принадлежит ли точка *A{x,* у) заданной фигуре.

Задача 2. Ввести два числа и символ — знак арифметической операции. В зависимости от введенного знака операции вычислить значение арифметического выражения.

Задача 3. Даны три числа: *а*, *Ь, с.* Определить, могут ли они быть сторонами треугольника, и если могут, то определить его тип: равносторонний, равнобедренный, разносторонний. (Условие существования треугольника: сумма длин любых двух сторон треугольника превышает длину 3-й стороны. Следует также учесть случай, когда длина одной из сторон равна нулю или имеет отрицательное значение.)

Задача 4. Ввести два целых числа. Вывести в порядке убывания все числа, лежащие между ними, и количество этих чисел. Каждое третье число не печатать и не учитывать.

Задача 5. В 1202 г. итальянский математик Леонард Фибоначчи подсчитывал, на сколько

увеличивается число кроликов в хозяйстве каждый год. При этом он получил последовательность такого вида: 1, 1, 2, 3, 5, 8, 13, 21, 34 ... . Написать программу, которая для заданного числа A выводит *N* членов последовательности Фибоначчи.

Оценка	Показатели оценки
3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен с одной ошибкой
5	Алгоритм составлен без ошибок

Задание №2

Задача 1. определить функцию сложности алгоритма по результатам эксперимента:

N	Количество перестановок
5	62

Задача 2. Определить функцию сложности алгоритма по результатам эксперимента:

N	Время работы, с
1000	0,134

Оценка	Показатели оценки
3	Решена одна задача
4	Решены обе задачи, в одной из них допушена не значительная ошибка
5	Обе задачи решены верно

Задание №3

Напишите инструкции по работе со средой программирования Visual Studio

- добавление файлов в созданный проект;

- выполнение отладки программы;

- выполнение программы по шагам.

Оценка	Показатели оценки
3	Написана одна из инструкций
4	Написано две инструкци
5	Написаны все инструкции

Задание №4

Оформите представленный код программы в соответствии со стандартом кодирования:

Вершины треугольника заданы массивом точек. Точки являются объектом класса CPoint. Написать функцию, вычисляющую площадь треугольника по формуле Герона:

#include #include class CPoint { private: int x, y; public:

CPoint(int _x, int *_y)*

(x = _x; y = _J/;}

CPoint()

{x = 0; y = 0; } void SetX(int _x)

{x - _x;}

void SetY(int _y)

(Y = _Y; > int GetX()

{ return x; } int GetY() { return y; } };

float length(CPoint pi, CPoint p2)

{

int xl, x2, yl, y2, L; xl = pi.GetX(); yl - pi.GetY(); x2 = p2.GetX(); y2 = p2.GetY();

L = sqrt(pow(x2-xl, 2.0) + pow(y2-yl, 2.0)); return L;

float square(CPoint *mas)

{float Ll, L2, L3, p;

LI = length(mas[0], mas[l]);

L2 = length(mas[1], mas[2]);

L3 = length(mas[2], mas[0]); p = (Ll + L2 + L3)/2.0; if (p*(p-Ll)*(p-L2)*(p-L3)>=0)

return sqrt(p*(p - Ll)*(p - L2)*(p - L3)); else (printf("Треугольник построить нельзя! "); return -1;

}} int main(){

CPoint mas [3]; int x, y;

for (int i = 0; i<3; i++){printf("First point:"); scanf("%d%d", &x, &y); mas [i] .SetX(x); mas[i].SetY(y);}

printf("S = %3.2f ", square(mas)); return 0;}

Оценка	Показатели оценки
3	программа оформлена частично в соответствии со стандартом кодирования
4	программа оформлена в соответствии со стандартом кодирования, с небольшими недочетами
5	программа оформлена полностью в соответствии со стандартом кодирования

Задание №5

Выполните проверку и отладку следующего кода программы

Написать класс «герой». Члены класса: имя героя, его возраст. Написать класс «параметры». Члены класса: сила, зашита, скорость. На базе этих двух классов написать класс «воин», включающий в себя все перечисленные члены.

Иерархия классов будет выглядеть следующим образом:

#include

#include

class Hero //начало определения 1-го базового класса Hero

{

protected;

```
char name [32 ] ; //защищенная член-переменная name класса Него

int age; //защищенная член-переменная аде класса Него

public:

Него () //определение конструктора класса Него без параметров

{

strcpy(name, "NoName");

//копирование в член-переменную name текста NoName аде = 0; //присвоить члену-переменной аде значения О

}

Hero(char* name, int age)

//определение конструктора класса Него с двумя параметрами

{

strcpy (this ->name, name) ; /*копирование члена-переменной

name (this->name,) значения параметра name V

this->age - age;

//присвоить члену-переменной age (this->age^ значение параметра age

}

void info () //определения функции-члена info

{

printf("Hero:%s, %d ", name, age);

//вывод на экран названия класса и значений членов-переменных name и аде

}

}; //конец определения базового класса Него

class Parametry //начало определения базового класса Parametry

{
```

```
protected:

int si la; //защищенная член-переменная si la класса Parametry

int zashita;

//защищенная член-переменная zashita класса Parametry

int skorost;

//защищенная член-переменная skorost класса Parametr

public:

Parametry(int _sila = 0, int _zashita = 0, int _skoros = 0)

//конструктор класса Parametry со значениями поумолчаниию

{

sila = _sila; zashita = _zashita; skorost - _skorost;

}

void info() //определение члена-функции info класса Parametry

printf("Parametry: %d, %d, %d ", sila, zashita,

skorost) ; /*вывод на экран названия класса и значении

членов-переменных sila, zashita, skorost*/

}

}; //конец определения класса Parametry

class Warrior : public Hero, Parametry

//начало определ. класса Warrior наследника классов Hero, Parametry { public:

/*определение конструктора класса Warrior инициализацией «родительских» конструкторов
классов Her o и Parametry*/

Warrior(char* _name, int _age, int _sila, int _zashita, int _skorost): Hero(_name, _age), Parametry(_sil,
_zashita, _skoros)

{ }
```

```
void info() //определение члена-функции info класса Warrior

{

printf("Warrior: ");

//вывод на экран названия класса Warrior Hero : : info () ; //вызов функции-члена info класса Hero

Parametry::info();

//вызов функции-члена info класса Parametry

}

}; //конец определения класса Warrior

int main()

Warrior w("Ivan", 30, 100, 300, 15); w.info(); return 0;

}
```

Оценка	Показатели оценки
3	проверка и отладка выполнена, программа работает с ошибками
4	проверка и отладка выполнена, программа работает с небольшими недочетами
5	проверка и отладка выполнена, программа работает верно