Министерство образования Иркутской области
Государственное бюджетное профессиональное образовательное учреждение Иркутской области
«Иркутский авиационный техникум»

УТВЕРЖДАЮ
Директор
ГБПОУИО «ИАТ»
_____ Якубовский А.Н.
«31» мая 2021 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

ОП.04 Основы алгоритмизации и программирования

специальности

09.02.07 Информационные системы и программирование

Иркутск, 2021

Рассмотрена
цикловой комиссией
КС, ИСП №9 от 25.05.2021 г.

Председатель ЦК

 /М.А. Кудрявцева /

№	Разработчик ФИО
1	Филимонова Ольга Николаевна

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Область применения фонда оценочных средств (ФОС)

ФОС по дисциплине является частью программы подготовки специалистов среднего звена по специальности 09.02.07 Информационные системы и программирование

1.2. Место дисциплины в структуре ППССЗ:

ОП.00 Общепрофессиональный цикл.

1.3. Цели и задачи дисциплины – требования к результатам освоения дисциплины

В результате освоения дисциплины обучающийся должен	№ дидактической единицы	Формируемая дидактическая единица
Знать	1.1	Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
	1.2	Эволюцию языков программирования, их классификацию, понятие системы программирования
	1.3	Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти
	1.4	Подпрограммы, составление библиотек подпрограмм
	1.5	Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения
Уметь	2.1	Разрабатывать алгоритмы для конкретных задач
	2.2	Использовать программы для графического отображения алгоритмов
	2.3	Определять сложность работы алгоритмов
	2.4	Работать в среде программирования

	2.5	Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования
	2.6	Оформлять код программы в соответствии со стандартом кодирования
	2.7	Выполнять проверку, отладку кода программы

1.4. Формируемые компетенции:

ОК.1 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК.2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК.4 Эффективно взаимодействовать и работать в коллективе и команде
ОК.5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК.9 Пользоваться профессиональной документацией на государственном и иностранном языках
ПК.5.4 Производить разработку модулей информационной системы в соответствии с техническим заданием

2. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ, ИСПОЛЬЗУЕМЫЙ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

2.1 Текущий контроль (ТК) № 1

Тема занятия: 2.1.1.Структурная организация данных.

Метод и форма контроля: Письменный опрос (Опрос)

Вид контроля: Самостоятельная работа

Дидактическая единица: 1.2 Эволюцию языков программирования, их классификацию, понятие системы программирования

Занятие(-я):

1.1.1.Обзор языков программирования. стандарты языков программирования.

1.1.2.Жизненный цикл программ. Основные этапы решения задач на компьютере.

1.1.3.Введение в язык С++. Правила оформления текстов программ.

1.1.5.Программный продукт и его характеристики.

Задание №1

Сопоставить даты создания языков программирования перейдя по ссылке: https://learningapps.org/131279 .

или по QR коду:

Оценка	*Показатели оценки*
3	10 правильных ответов.
4	14 правильных ответов.
5	18 правильных ответов.

Дидактическая единица: 1.3 Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти

Занятие(-я):

1.1.4.Структура программы на языке С++.

Задание №1

Выполните тест, в личном кабинете - "Введение в язык С++".

Оценка	*Показатели оценки*
3	3 правильных ответа;
4	4-5 правильных ответов;
5	6 правильных ответов.

2.2 Текущий контроль (ТК) № 2

Тема занятия: 2.1.13.Решение задач.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с использованием ИКТ
Дидактическая единица: 1.1 Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
Занятие(-я):
2.1.3.Модели объектов и процессов.
2.1.4.Модели объектов и процессов.
2.1.5.Составление линейных и разветвленных алгоритмов.
Задание №1
Выполните тест "Понятие алгоритм", в личном кабинете.

Оценка	*Показатели оценки*
3	5 правильных ответов;
4	6-8 правильных ответов ;
5	9 правильных ответов;

Дидактическая единица: 2.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования
Занятие(-я):
2.1.8.Программирование разветвленных алгоритмов.
2.1.9.Программирование разветвленных алгоритмов.
2.1.10.Программирование разветвленных алгоритмов.
2.1.11.Программирование разветвленных алгоритмов. Условная операция. Оператор выбора.
2.1.12.Решение задач.
Задание №1
Написать погрaмму для построенных алгоритмов (Задание 2) на языке програмирования C++.

Оценка	*Показатели оценки*
3	Программа написана для линейного алгоритма.
4	Программа написана для линейного и разветвляющегося алгоритмов, но допущены незначительные ошибки.
5	Программа написана для линейного и разветвляющегося алгоритмов без ошибок.

Дидактическая единица: 2.6 Оформлять код программы в соответствии со

стандартом кодирования

Занятие(-я):

2.1.8.Программирование разветвленных алгоритмов.

2.1.9.Программирование разветвленных алгоритмов.

2.1.10.Программирование разветвленных алгоритмов.

2.1.11.Программирование разветвленных алгоритмов. Условная операция. Оператор выбора.

2.1.12.Решение задач.

Задание №1

Оформите код написанных программ в соответствии со стандартом кодирования.

Оценка	***Показатели оценки***
3	Код оформлен без соблюдения правил.
4	Код программы частично оформлен в соответствии со стандартом.
5	Код программы оформлен в соответствии со стандартом.

Дидактическая единица: 2.1 Разрабатывать алгоритмы для конкретных задач

Занятие(-я):

2.1.3.Модели объектов и процессов.

2.1.4.Модели объектов и процессов.

2.1.5.Составление линейных и разветвленных алгоритмов.

Задание №1

Построить алгоритм решения задач в виде блок-схем:

(*представлен один из вариантов задач)*

1. Поменять местами содержимое переменных A и B и вывести новые значения A и B.

2. Для данного вещественного x найти значение следующей функции f, принимающей вещественные значения:

$$f(x) = \begin{cases} 2\sin(x), \text{если } x > 0, \\ 6 - x, \text{если } x \leq 0. \end{cases}$$

Оценка	***Показатели оценки***
3	Алгоритм построен для одной задачи;
4	Алгоритм построен для двух задач, допущены ошибки в построении блок-схемы;

5	Алгоритм построен верно для всез задач;

2.3 Текущий контроль (ТК) № 3

Тема занятия: 2.1.21.Решение задач.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с применением ИКТ
Дидактическая единица: 2.2 Использовать программы для графического отображения алгоритмов
Занятие(-я):
2.1.5.Составление линейных и разветвленных алгоритмов.
2.1.14.Решение задач на составление циклических алгоритмов.
2.1.15.Решение задач на составление циклических алгоритмов.
Задание №1
1. Построить алгоритм решения следующей задачи в виде блок схемы:
(Представлен один из вариантов задачи)
Дана числовая последовательность из Т элементов. Вывести номера всех нулевых элементов.
2. Оформить схему алгоритма в программе для графического отображения алгоритмов.

Оценка	***Показатели оценки***
3	Алгоритм постоен, схема создана в графическом редакторе Paint.
4	Алгоритм построен, схема создана в текстовом редакторе, средствами рисования.
5	Алгоритм постоен, схема создана в специализированной программе или в он-лайн сервисе.

Дидактическая единица: 2.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования
Занятие(-я):
2.1.13.Решение задач.
2.1.14.Решение задач на составление циклических алгоритмов.
2.1.15.Решение задач на составление циклических алгоритмов.
2.1.16.Программирование циклических алгоритмов.
2.1.17.Программирование циклических алгоритмов.
2.1.18.Программирование циклических алгоритмов.
2.1.19.Программирование циклических алгоритмов.
2.1.20.Решение задач.
Задание №1
Напишите программу для построенного алгоритма на языке программирования C++.

Оценка	*Показатели оценки*
3	Программа написана, но не работает из-за ошибок.
4	Программа написана, работает с незначительными ошибками.
5	Программа работает без ошибок.

Дидактическая единица: 2.6 Оформлять код программы в соответствии со стандартом кодирования

Занятие(-я):

2.1.13.Решение задач.
2.1.16.Программирование циклических алгоритмов.
2.1.17.Программирование циклических алгоритмов.
2.1.18.Программирование циклических алгоритмов.
2.1.19.Программирование циклических алгоритмов.
2.1.20.Решение задач.

Задание №1

Оформите код написанных программ в соответствии со стандартом кодирования.

Оценка	*Показатели оценки*
3	Код оформлен без соблюдения правил.
4	Код программы частично оформлен в соответствии со стандартом.
5	Код программы оформлен в соответствии со стандартом.

Дидактическая единица: 2.7 Выполнять проверку, отладку кода программы

Занятие(-я):

2.1.8.Программирование разветвленных алгоритмов.
2.1.9.Программирование разветвленных алгоритмов.
2.1.10.Программирование разветвленных алгоритмов.
2.1.11.Программирование разветвленных алгоритмов. Условная операция. Оператор выбора.
2.1.12.Решение задач.
2.1.13.Решение задач.
2.1.16.Программирование циклических алгоритмов.
2.1.17.Программирование циклических алгоритмов.
2.1.18.Программирование циклических алгоритмов.
2.1.19.Программирование циклических алгоритмов.
2.1.20.Решение задач.

Задание №1

Выполните проверку и отладку программа.

Оценка	*Показатели оценки*
3	Программа не выполняет условие цикла.
4	В прорамме не предусмотрено дно из условий цикла.
5	В программе предусмотренно различные вариации решения.

2.4 Текущий контроль (ТК) № 4

Тема занятия: 2.1.37.Решение задач.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с применением ИКТ
Дидактическая единица: 1.3 Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти
Занятие(-я):
2.1.1.Структурная организация данных.
2.1.2.Структурная организация данных.
2.1.6.Основы работы в интегрированной среде разработки MS Visual Studio.
2.1.7.Программирование задач линейной структуры.
2.1.13.Решение задач.
2.1.14.Решение задач на составление циклических алгоритмов.
2.1.15.Решение задач на составление циклических алгоритмов.
2.1.16.Программирование циклических алгоритмов.
2.1.21.Решение задач.
2.1.23.Препроцессорные средства.
2.1.24.Память. Адреса. Указатели.
2.1.25.Одномерные массивы.
2.1.26.Методы сортировки.
2.1.27.Методы поиска.
2.1.32.Работа со строками.
2.1.36.Решение задач.
Задание №1
Ответьте на вопросы теста "Массивы", в информационной аналитической системе техникума.

Оценка	*Показатели оценки*
3	50% правильных ответов;
4	от 70% до 90% правильных ответов;
5	100% правильных ответов;

Дидактическая единица: 2.3 Определять сложность работы алгоритмов

Занятие(-я):

2.1.22.Функция сложности алгоритма.

2.1.25.Одномерные массивы.

Задание №1

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.

Дидактическая единица: 2.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования

Занятие(-я):

2.1.21.Решение задач.

2.1.28.Решение задач с использованием одномерных массивов.

2.1.29.Решение задач с использованием одномерных массивов.

2.1.30.Решение задач с использованием одномерных массивов.

2.1.31.Решение задач с использованием одномерных массивов.

2.1.33.Решение задач со строками.

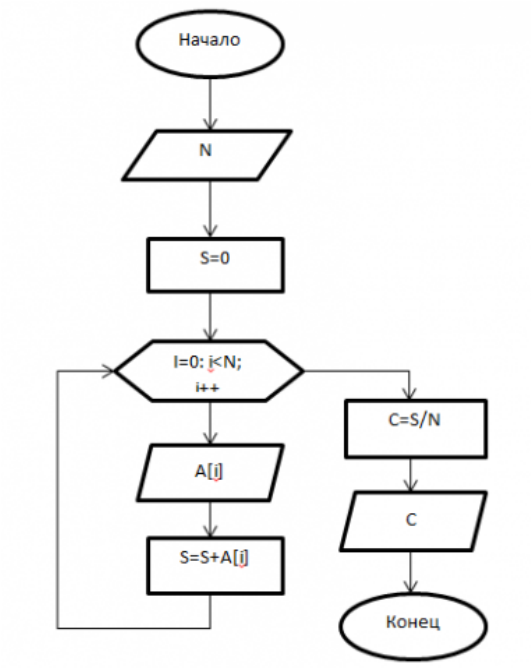
2.1.34.Решение задач со строками.

2.1.35.Решение задач со строками.

2.1.36.Решение задач.

Задание №1

Для данного алгоритма написать программу на языке C++.



Оценка	*Показатели оценки*
3	программа написана не значительными с ошибками;
4	программа написана, работает с неблышими недочетами;
5	Программа написана, работает правильно;

Дидактическая единица: 2.4 Работать в среде программирования
Занятие(-я):
2.1.6.Основы работы в интегрированной среде разработки MS Visual Studio.
2.1.7.Программирование задач линейной структуры.
Задание №1
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.

2.5 Текущий контроль (ТК) № 5

Тема занятия: 2.2.6.Решение задач.

Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с применением ИКТ
Дидактическая единица: 2.1 Разрабатывать алгоритмы для конкретных задач
Занятие(-я):
2.1.13.Решение задач.
2.1.14.Решение задач на составление циклических алгоритмов.
2.1.15.Решение задач на составление циклических алгоритмов.
2.1.25.Одномерные массивы.
2.1.26.Методы сортировки.
2.1.27.Методы поиска.
2.1.32.Работа со строками.
Задание №1
Разработать алгоритм для следующей задачи:
(Пример одного из вариантов)
Сформировать матрицу A{6,7}. вывести ее на экран. Найти произведение элементов всех строк.

Оценка	***Показатели оценки***
3	Алгоритм разработан с ошибками.
4	Алгоритм составлен с не значительными недочетами.
5	Алгоритм составлен верно.

Дидактическая единица: 2.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования
Занятие(-я):
2.1.37.Решение задач.
2.2.2.Решение задач с использованием двумерных массивов.
2.2.3.Решение задач с использованием двумерных массивов.
2.2.4.Решение задач с использованием двумерных массивов.
Задание №1
Написать программу для составленного алгоритма на языке программирования C++.

Оценка	***Показатели оценки***
3	Программа написана, с не значительными с ошибками.
4	Программа написана, работает с небльшими недочетами.
5	Программа написана, работает правильно.

Дидактическая единица: 2.6 Оформлять код программы в соответствии со стандартом кодирования

Занятие(-я):
2.1.21.Решение задач.
2.1.23.Препроцессорные средства.
2.1.24.Память. Адреса. Указатели.
2.1.28.Решение задач с использованием одномерных массивов.
2.1.29.Решение задач с использованием одномерных массивов.
2.1.30.Решение задач с использованием одномерных массивов.
2.1.31.Решение задач с использованием одномерных массивов.
2.1.33.Решение задач со строками.
2.1.34.Решение задач со строками.
2.1.35.Решение задач со строками.
2.1.36.Решение задач.
2.1.37.Решение задач.
2.2.2.Решение задач с использованием двумерных массивов.
2.2.3.Решение задач с использованием двумерных массивов.
2.2.4.Решение задач с использованием двумерных массивов.
2.2.5.Решение задач.

Задание №1

Оформите код написанных программ в соответствии со стандартом кодирования.

Оценка	***Показатели оценки***
3	Код оформлен без соблюдения правил.
4	Код программы частично оформлен в соответствии со стандартом.
5	Код программы оформлен в соответствии со стандартом.

Дидактическая единица: 2.7 Выполнять проверку, отладку кода программы

Занятие(-я):
2.1.28.Решение задач с использованием одномерных массивов.
2.1.29.Решение задач с использованием одномерных массивов.
2.1.30.Решение задач с использованием одномерных массивов.
2.1.31.Решение задач с использованием одномерных массивов.
2.1.33.Решение задач со строками.
2.1.34.Решение задач со строками.
2.1.35.Решение задач со строками.
2.1.36.Решение задач.
2.1.37.Решение задач.
2.2.2.Решение задач с использованием двумерных массивов.
2.2.3.Решение задач с использованием двумерных массивов.
2.2.4.Решение задач с использованием двумерных массивов.

Задание №1
Выполните отладу программы и пошаговое выпонение. Результаты пошагового выполнения представьте в виде скриншотов.

Оценка	*Показатели оценки*
3	Выполнена отладка программы.
4	программа выполнена пошагово, скриншоты представлены не все.
5	программа выполнена пошагово, представлены все скриншоты.

2.6 Текущий контроль (ТК) № 6

Тема занятия: 2.2.19.Решение задач.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с применением ИКТ
Дидактическая единица: 1.3 Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти
Занятие(-я):
2.1.37.Решение задач.
2.2.1.Двумерные массивы (матрицы).
2.2.5.Решение задач.
2.2.6.Решение задач.
2.2.7.Понятие функции.
2.2.8.Использование массивов в качестве параметров.
2.2.9.Итеративные и рекрусивные алгоритмы.
2.2.10.Итеративные и рекрусивные алгоритмы.
2.2.14.Решение задач с использованием переменных комбинированного типа.
2.2.15.Динамические структуры данных (списки) Формирование списков.
Задание №1
Ответьте на вопросы теста "Двумерные массивы. Работа со стркутурами" в информационно аналитической системе техникума.

Оценка	*Показатели оценки*
3	50-74 % правильных ответов;
4	75-98 % правильных ответов;
5	99-100 % правильных ответов;

Дидактическая единица: 2.1 Разрабатывать алгоритмы для конкретных задач
Занятие(-я):

2.2.11.Основные определения теории графов.

Задание №1

Построить алгоритм для решения следующей задачи:

(представлен один из вариантов)

$$a_n = \frac{n!}{2^n}$$

Оценка	*Показатели оценки*
3	Алгоритм постоен, схема создана в графическом редакторе Paint.
4	Алгоритм построен, схема создана в текстовом редакторе, средствами рисования.
5	Алгоритм постоен, схема создана в специализированной программе или в он-лайн сервисе.

Дидактическая единица: 2.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования

Занятие(-я):

2.2.16.Решение задач со структурами.

Задание №1

Написать программу по составленному алгоритму на языке программирования C++. Вычисления организовать в виде рекруссивной функции. Программу выполнить по шагам, записать последовательное изменение стека.

Оценка	*Показатели оценки*
3	Программа написана, но не работает из-за ошибок.
4	Программа написана, работает с незначительными ошибками.
5	Программа работает без ошибок.

Дидактическая единица: 2.6 Оформлять код программы в соответствии со стандартом кодирования

Занятие(-я):

2.2.6.Решение задач.

2.2.12.Решение задач с применением рекрусивных функций.

2.2.13.Решение задач с применением рекрусивных функций.

2.2.16.Решение задач со структурами.

2.2.17.Решение задач со структурами.

2.2.18.Решение задач.

Задание №1
Оформите код написанных программ в соответствии со стандартом кодирования.

Оценка	*Показатели оценки*
3	Код оформлен без соблюдения правил.
4	Код программы частично оформлен в соответствии со стандартом.
5	Код программы оформлен в соответствии со стандартом.

2.7 Текущий контроль (ТК) № 7

Тема занятия: 3.1.12.Решение задач.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с применением ИКТ
Дидактическая единица: 1.4 Подпрограммы, составление библиотек подпрограмм
Занятие(-я):
2.2.7.Понятие функции.
2.2.8.Использование массивов в качестве параметров.
2.2.9.Итеративные и рекрусивные алгоритмы.
2.2.10.Итеративные и рекрусивные алгоритмы.
2.2.18.Решение задач.
2.2.19.Решение задач.
Задание №1
Дайте ответы на следующие вопросы:

1. напишите структуру функции в общем виде и объясните, для чего используется каждый раздел функции. Приведите примеры определения функции;
2. что такое передача по значению, ссылке, указателю? объясните на примерах;
3. что произойдет, еслиглобальная и локальная переменные имеют одинаковые имена? Объясните на примерах.

Оценка	*Показатели оценки*
3	Даны ответы на 2 вопроса, без объяснений на примерах.
4	Даны ответы на три вопроса, без объяснений на примерах.
5	Даны ответы на все вопросы, даны объяснения на примерах.

Дидактическая единица: 1.5 Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и

переопределения

Занятие(-я):

3.1.1.Объектно-ориентированная модель. Этапы разработки программных продуктов с использованием ООП.

3.1.2.Классы. Создание объектов (экземпляров) класса.

3.1.3.Написание классов.

3.1.4.Особенности классов.

3.1.5.Наследование, полиморфизм.

3.1.8.Потоковый ввод/вывод.

Задание №1

Дайте ответы на вопросы теста:

1. Что означает аббревиатура ООП:
а) объектный образ в программировании;
б) объектно-ориентированное программирование;
в) объективно ориентированное программирование.

2. Принцип инкапсуляции обеспечивает:
а) объединение данных и методов работы с ними в классе;
б) доступ к членам класса;
в) сокрытие данных внутри класса.

3. Укажите правильный вариант определения класса в программе:
а) class Test {
public: int a;
};
б) class Test {
private: float x;
}
в) оба варианта правильные.

4. Спецификатор доступа private обеспечивает:
а) доступность членов класса в методах данного класса и в дружественных функциях данного класса;
б) доступность членов класса в методах данного класса, в дружественных функциях данного класса и в методах наследников данного класса;
в) доступность членов класса во всех функциях программы.

5. Спецификатор доступа protected обеспечивает:
а) доступность членов класса в методах данного класса и в дружественных функциях данного класса;

б) доступность членов класса в методах данного класса, в дружественных функциях данного класса и в методах наследников данного класса;
в) доступность членов класса во всех функциях программы.

6. Спецификатор доступа public обеспечивает:
а) доступность членов класса в методах данного класса и в дружественных функциях данного класса;
б) доступность членов класса в методах данного класса, в дружественных функциях данного класса и в методах наследников данного класса;
в) доступность членов класса во всех функциях программы.

7. Что будет выведено на экран в результате выполнения следующей программы?

```
class CLight
{
int a;
};
...
CLight L;
L.a = 45;
printf("a = %d\n ", L.a);
...
```

а) a = 45;
б) программа не запустится, так как доступ к полю «a» необходимо получить, используя операцию: L->a = 45;
в) программа не запустится, так как переменная «a» является закрытой.

8. Конструктор — это:
а) специальный метод класса с тем же именем, что и сам класс;
б) специальный метод класса, не имеющий параметров и не возвращающий никакого значения;
в) механизм создания новых объектов класса.

9. В классе может быть только:
а) единственный конструктор;
б) два конструктора — по умолчанию и с параметрами;
в) произвольное количество конструкторов.

10. Деструктор — это:
а) специальный метод класса с тем же именем, что и сам класс с префиксом — тильдой (~);
б) автоматически создаваемый метод класса, предназначенный для удаления

объектов класса;
в) механизм разрушения объектов класса.

Оценка	*Показатели оценки*
3	5-6 правильных ответов;
4	7-9 правильных ответов;
5	10 правильных ответов;

Дидактическая единица: 2.6 Оформлять код программы в соответствии со стандартом кодирования

Занятие(-я):

2.2.19.Решение задач.
3.1.3.Написание классов.
3.1.6.Создание иерархии классов.
3.1.7.Создание иерархии классов.
3.1.9.Ввод/вывод различных типов данных.
3.1.10.Ввод/вывод различных типов данных.
3.1.11.Решение задач.

Задание №1

Напишите базовый класс Name, в котором храниться имя героя игры. На основе этого класса создать классы игроков: Warrior - воин, характеризуется именем и силой. Horse - лошадь, характеризуется именем и скоростью. На базе классов Warrior и Horse создать новый класс игрока: Centaur(Кентавр), который характеризуется именем, силой и скоростью. Вывести на экранразмеры созданных классов. Продемонстрировать разницу в классе Centaur при использовании абстрактных базовых классов. Информацию о классе записывается в файл.

Оценка	*Показатели оценки*
3	Созданы класы, отсутствуют характеристики игроков, на одном из уровней. Информация о классе не записывается в файл.
4	Созданы классы, даны характеристики игроков. Информация о классе не записывается в файл.
5	Созданы классы, даны характеристики игроков. Информация о классе записывается в файл.

3. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ, ИСПОЛЬЗУЕМЫЙ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

№ семестра	Вид промежуточной аттестации
4	Экзамен

Экзамен может быть выставлен автоматически по результатам текущих контролей
Текущий контроль №1
Текущий контроль №2
Текущий контроль №3
Текущий контроль №4
Текущий контроль №5
Текущий контроль №6
Текущий контроль №7

Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: ответьте на 15 вопросов теста и выполните одно практическое задание
Дидактическая единица для контроля:
1.1 Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции

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

Дидактическая единица для контроля:
1.2 Эволюцию языков программирования, их классификацию, понятие системы программирования

Задание №1

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

Дидактическая единица для контроля:
1.3 Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти
Задание №1
Вопрос 1. Выберите видовые отличия термина "Язык программирования"

- предназначен для записи программ;
- имеет определенный синтаксис;
- предназначен для передачи информации;
- машиноориентированный, низкого уровня.

Вопрос 2 Выберите видовые отличия термина "Алгоритм"

- ориентирована на пользователя;
- выполняется за конечное число шагов;

- предназначен для записи программ;
- имеет определенный синтаксис.

Вопрос 3. Выберите правильное определение для термина "Математическая модель"

- модель, описывающая математическими формулами соотношения между количественными характеристиками объекта моделирования
- объект, который воспроизводит наиболее характерные свойства реального процесса или устройства или концепции
- модель, которая определяет уровни взаимодействия систем и их стандартные названия и функции

Вопрос 4 Выберите родовое понятие для термина "Моделирование"

- система познания;
- набор объектов;
- множество методов.

Вопрос 5 Соотнесите термины и родовые понятия

алфавит		множество
идентификатор		набор символов
константа		величина
ключевые слова		единица языка

Вопрос 6 Соотнесите термины и видовые отличия

идентификатор		является именем объектов программы
константа		имеет постоянное значение
ключевые слова		имеют специальное значение для компилятора

Вопрос 7 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 8 Выберите пример правильного идентификатора в языке C++

- Fr_5;
- 10Sd;
- scanf.

Вопрос 9 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 10 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 11 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 5-7 вопросов
4	Даны правильные ответы на 8-10 вопросов
5	Даны правильные ответы на 11 вопросов

Дидактическая единица для контроля:
1.4 Подпрограммы, составление библиотек подпрограмм

Задание №1

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;}
- 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

Дидактическая единица для контроля:
1.5 Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения

Задание №1

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)
  {
  • • •
  }
- 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

Дидактическая единица для контроля:

2.1 Разрабатывать алгоритмы для конкретных задач

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 А 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

Дидактическая единица для контроля:

2.2 Использовать программы для графического отображения алгоритмов

Задание №1

Разработайте алгоритм для одной из задачи с использованием программы для графического отображения алгоритмов:

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 А 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

Дидактическая единица для контроля:

2.3 Определять сложность работы алгоритмов

Задание №1 (из текущего контроля)

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.

Дидактическая единица для контроля:

2.4 Работать в среде программирования

Задание №1 (из текущего контроля)

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.

Дидактическая единица для контроля:

2.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования

Задание №1

Написать программу для задачи, по приведенной схеме алгоритма:

Задача 1. Дан массив K из 13 элементов и целое число N (вводится с клавиатуры).

Из элементов массива К, больших числа Ы, сформировать массив С. Вывести оба массива на экран.

Задача 2. Найти сумму положиельных элементов двумерного массива.

Оценка	***Показатели оценки***
3	для представленного алгоритма программа написана, допущены ошибки
4	для представленного алгоритма программа написана, не предусмотрен один из критериев
5	для представленного алгоритма программа написана, учтены все из вариантов решения

Дидактическая единица для контроля:

2.6 Оформлять код программы в соответствии со стандартом кодирования

Задание №1

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

Дидактическая единица для контроля:

2.7 Выполнять проверку, отладку кода программы

Задание №1

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 Него
{
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