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

УТВЕРЖДАЮ
Директор
ГБПОУИО «ИАТ»
_____ Якубовский А.Н.
«08» февраля 2023 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

ОП.04 Основы алгоритмизации и программирования

специальности

09.02.07 Информационные системы и программирование

Иркутск, 2023

Рассмотрена
цикловой комиссией
ИСП протокол №9 от
17.05.2023 г.

№	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

Результаты освоения дисциплины	№ результата	Формируемый результат
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
Личностные результаты реализации программы воспитания	4.1	Демонстрирующий приверженность традиционным духовно-нравственным ценностям, культуре народов России, принципам честности, порядочности, открытости. Действующий и оценивающий свое поведение и поступки, поведение и поступки других людей с позиций традиционных российских духовно-нравственных, социокультурных ценностей и норм с учетом осознания последствий поступков. Готовый к деловому взаимодействию и неформальному общению с представителями разных народов, национальностей, вероисповеданий, отличающий их от участников групп с деструктивным и девиантным поведением. Демонстрирующий неприятие социально опасного поведения окружающих и предупреждающий его. Проявляющий уважение к людям старшего поколения, готовность к участию в социальной поддержке нуждающихся в ней
	4.2	Проявляющий и демонстрирующий уважение к труду человека, осознающий ценность собственного труда и труда других людей. Экономически активный, ориентированный на осознанный выбор сферы профессиональной деятельности с учетом личных жизненных планов, потребностей своей семьи, российского общества. Выражающий осознанную готовность к получению профессионального образования, к непрерывному образованию в течение жизни Демонстрирующий позитивное отношение к регулированию трудовых отношений. Ориентированный на самообразование и профессиональную переподготовку в условиях смены технологического уклада и сопутствующих социальных перемен. Стремящийся к формированию в сетевой среде личностно и профессионального конструктивного «цифрового следа»

	4.3	Осознающий и деятельно выражающий приоритетную ценность каждой человеческой жизни, уважающий достоинство личности каждого человека, собственную и чужую уникальность, свободу мировоззренческого выбора, самоопределения. Проявляющий бережливое и чуткое отношение к религиозной принадлежности каждого человека, предупредительный в отношении выражения прав и законных интересов других людей
	4.4	Бережливо относящийся к природному наследию страны и мира, проявляющий сформированность экологической культуры на основе понимания влияния социальных, экономических и профессионально-производственных процессов на окружающую среду. Выражающий деятельное неприятие действий, приносящих вред природе, распознающий опасности среды обитания, предупреждающий рискованное поведение других граждан, популяризирующий способы сохранения памятников природы страны, региона, территории, поселения, включенный в общественные инициативы, направленные на заботу о них
	4.5	Демонстрирующий умение эффективно взаимодействовать в команде, вести диалог, в том числе с использованием средств коммуникации
	4.6	Демонстрирующий навыки анализа и интерпретации информации из различных источников с учетом нормативно-правовых норм
	4.7	Демонстрирующий готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности

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 (45 минут)

Тема занятия: 1.1.11.Разветвленный алгоритм. 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):
1.1.4.Составление линейных и разветвленных алгоритмов.
1.1.5.Условный оператор if (полная и неполная формы).
1.1.7.Оператор выбора switch.

Задание №1 (5 минут)

Ответить на вопросы:

1. Выберите видовые отличия, которые относятся к понятию «Линейный алгоритм»

1. действия выполняются однократно;
2. действия выполняются в заданном порядке;
3. содержит одно или несколько логических условий;
4. содержит несколько ветвей вычислений.

2. Выберите видовые отличия, которые относятся к понятию «Разветвляющийся алгоритм»

1. действия выполняются однократно;
2. действия выполняются в заданном порядке;
3. содержит одно или несколько логических условий;
4. содержит несколько ветвей вычислений.

3. Выберите видовые отличия термина "Алгоритм"

1. Ориентирована на пользователя;
2. Выполняется за конечное число шагов;
3. Предназначен для записи программ;
4. Имеет определенный синтаксис.

Оценка	*Показатели оценки*
5	3 правильных ответов;
4	2 правильных ответов;
3	1 правильный ответов;

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

Занятие(-я):

1.1.1.История языка программирования. Основные понятия.

1.1.2.Структура программы. Типы данных языка программирования.

1.1.3.Операторы ввода и вывода.

Задание №1 (7 минут)

Ответить на вопросы:

1. Выберите видовые отличия термина "Язык программирования"

1. предназначен для записи программ;
2. имеет определенный синтаксис;
3. предназначен для передачи информации;
4. машиноориентированный, низкого уровня.

2. Выберите родовое понятие термина «Жизненный цикл»

1. описание изменений;
2. определенная последовательность состояний;
3. изменения в течении некоторого периода;
4. фиксация изменений.

3. Выберите правильное определение для термина "Математическая модель"

1. модель, описывающая математическими формулами соотношения между количественными характеристиками объекта моделирования;
2. объект, который воспроизводит наиболее характерные свойства реального процесса или устройства или концепции;
3. модель, которая определяет уровни взаимодействия систем и их стандартные

названия и функции.

4. Выберите родовое понятие для термина "Моделирование"

1. Система познания;
2. Набор объектов;
3. Множество методов.

Оценка	*Показатели оценки*
5	4 правильных ответов;
4	3 правильных ответов;
3	2 правильный ответ;

Дидактическая единица: 2.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования
Занятие(-я):
1.1.7.Оператор выбора switch.
1.1.8.Решение задач с оператором выбора switch.
1.1.9.Создание проекта “Текстовый квест” (Создание игры "Угадай число").
1.1.10.Разветвленный алгоритм. Решение задач.
Задание №1 (15 минут)
Написать программу для задачи *(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:

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

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):

1.1.3.Операторы ввода и вывода.

1.1.7.Оператор выбора switch.

1.1.8.Решение задач с оператором выбора switch.

1.1.9.Создание проекта “Текстовый квест” (Создание игры "Угадай число").

1.1.10.Разветвленный алгоритм. Решение задач.

Задание №1 (5 минут)

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):

1.1.4.Составление линейных и разветвленных алгоритмов.

1.1.5.Условный оператор if (полная и неполная формы).

Задание №1 (13 минут)

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:

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

Оценка	***Показатели оценки***
3	Алгоритм построен с грубыми ошибками (не правильное использование блоков);
4	Алгоритм построен, допущены не значительные ошибки в построении блок-схемы;
5	Алгоритм построен верно;

2.2 Текущий контроль (ТК) № 2 (45 минут)

Тема занятия: 1.1.17.Циклический алгоритм. 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):
1.1.4.Составление линейных и разветвленных алгоритмов.
1.1.12.Цикл с параметром for.
1.1.14.Цикл с предусловием while, цикл с постусловием do-while.
Задание №1 (15 минут)
1. Построить алгоритм решения следующей задачи в виде блок схемы *(Представлен один из вариантов задачи):*
Дана числовая последовательность из T элементов. Вывести номера всех нулевых элементов.
2. Оформить схему алгоритма в программе для графического отображения алгоритмов.

Оценка	***Показатели оценки***
3	Алгоритм построен, схема создана в графическом редакторе Paint.
4	Алгоритм построен, схема создана в текстовом редакторе, средствами рисования.
5	Алгоритм построен, 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):
1.1.11.Разветвленный алгоритм. Решение задач.
1.1.12.Цикл с параметром for.
1.1.13.Применение цикла с параметром.
1.1.14.Цикл с предусловием while, цикл с постусловием do-while.
1.1.15.Применение циклов с предусловием и с постусловием.
1.1.16.Циклический алгоритм. Решение задач.
Задание №1 (20 минут)
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):

1.1.11.Разветвленный алгоритм. Решение задач.

1.1.13.Применение цикла с параметром.

1.1.15.Применение циклов с предусловием и с постусловием.

1.1.16.Циклический алгоритм. Решение задач.

Задание №1 (5 минут)

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):

1.1.7.Оператор выбора switch.

1.1.8.Решение задач с оператором выбора switch.

1.1.9.Создание проекта “Текстовый квест” (Создание игры "Угадай число").

1.1.10.Разветвленный алгоритм. Решение задач.

1.1.11.Разветвленный алгоритм. Решение задач.

1.1.13.Применение цикла с параметром.

1.1.15.Применение циклов с предусловием и с постусловием.

1.1.16.Циклический алгоритм. Решение задач.

Задание №1 (5 минут)

Выполните проверку и отладку программы.

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

2.3 Текущий контроль (ТК) № 3 (45 минут)

Тема занятия: 1.1.26.Одномерные массивы. 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):
1.1.2.Структура программы. Типы данных языка программирования.
1.1.3.Операторы ввода и вывода.
1.1.6.Решение задач с условным оператором if.
1.1.11.Разветвленный алгоритм. Решение задач.
1.1.12.Цикл с параметром for.
1.1.13.Применение цикла с параметром.
1.1.14.Цикл с предусловием while, цикл с постусловием do-while.
1.1.17.Циклический алгоритм. Решение задач.
1.1.18.Препроцессорные средства.
1.1.19.Память. Адреса. Указатели.
1.1.20.Одномерные массивы. Динамические одномерные массивы.
1.1.23.Обработка элементов массива: методы поиска, методы сортировки.
1.1.25.Одномерные массивы. Решение задач.
Задание №1 (10 минут)
Ответьте на вопросы теста "Массивы", в информационно-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):
1.1.20.Одномерные массивы. Динамические одномерные массивы.
1.1.23.Обработка элементов массива: методы поиска, методы сортировки.
Задание №1 (5 минут)
Задача 1. Определить функцию сложности алгоритма по результатам эксперимента:

N	Количество перестановок
5	62

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

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

Занятие(-я):

1.1.17.Циклический алгоритм. Решение задач.

1.1.21.Решение задач с использованием одномерных массивов.

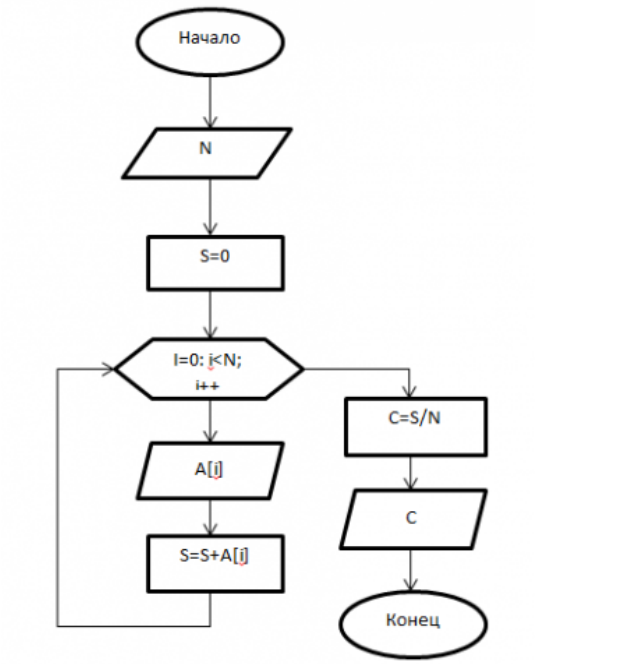
1.1.22.Решение задач с использованием одномерных массивов.

1.1.24.Решение задач на обработку элементов массива.

1.1.25.Одномерные массивы. Решение задач.

Задание №1 (10 минут)

Для данного алгоритма написать программу на языке С++.



Оценка	*Показатели оценки*
3	Программа написана с незначительными ошибками.
4	Программа написана, работает с небольшими недочетами.
5	Программа написана, работает правильно.

Дидактическая единица: 2.4 Работать в среде программирования
Занятие(-я):
1.1.5.Условный оператор if (полная и неполная формы).
1.1.6.Решение задач с условным оператором if.
Задание №1 (20 минут)
В среде разработки Visual Studio напишите программу, которая вычисляет сумму первых n целых положительных четных чисел. Количество суммируемых чисел вводится во время работы программы.
Рекомендуемый вид экрана:

```
?????????? ????? ???????????? ?????? ?????.??????? ????????
?? ?????????? ?????:12????? ?????? 12 ???????????? ??????
????? ????? 156.
```

(*представлен один из вариантов задач)*

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

2.4 Текущий контроль (ТК) № 4 (45 минут)
Тема занятия: 1.2.6.Двумерные массивы. 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):
1.1.11.Разветвленный алгоритм. Решение задач.
1.1.12.Цикл с параметром for.
1.1.14.Цикл с предусловием while, цикл с постусловием do-while.
1.1.20.Одномерные массивы. Динамические одномерные массивы.
1.1.23.Обработка элементов массива: методы поиска, методы сортировки.
1.1.27.Способы обработки элементов массива.
Задание №1 (15 минут)
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):

1.1.26.Одномерные массивы. Решение задач.
1.1.27.Способы обработки элементов массива.
1.2.2.Решение задач с использованием двумерных массивов.
1.2.3.Решение задач с использованием двумерных массивов.
1.2.4.Решение задач с использованием двумерных массивов.

Задание №1 (20 минут)

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 небольшими недочетами.
5	Программа написана, работает правильно.

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

Занятие(-я):

1.1.17.Циклический алгоритм. Решение задач.
1.1.18.Препроцессорные средства.
1.1.19.Память. Адреса. Указатели.
1.1.21.Решение задач с использованием одномерных массивов.
1.1.22.Решение задач с использованием одномерных массивов.
1.1.24.Решение задач на обработку элементов массива.
1.1.25.Одномерные массивы. Решение задач.
1.1.26.Одномерные массивы. Решение задач.
1.2.2.Решение задач с использованием двумерных массивов.
1.2.3.Решение задач с использованием двумерных массивов.
1.2.4.Решение задач с использованием двумерных массивов.
1.2.5.Двумерные массивы. Решение задач.

Задание №1 (5 минут)

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):
1.1.21.Решение задач с использованием одномерных массивов.
1.1.22.Решение задач с использованием одномерных массивов.
1.1.24.Решение задач на обработку элементов массива.
1.1.25.Одномерные массивы. Решение задач.
1.1.26.Одномерные массивы. Решение задач.
1.2.2.Решение задач с использованием двумерных массивов.
1.2.3.Решение задач с использованием двумерных массивов.
1.2.4.Решение задач с использованием двумерных массивов.
Задание №1 (5 минут)
Выполните отладку программы и пошаговое выполнение. 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.

2.5 Текущий контроль (ТК) № 5 (45 минут)

Тема занятия: 1.2.13.Решение задач с применением рекрусивных функций.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Отчет
Дидактическая единица: 1.1 Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
Занятие(-я):
1.1.12.Цикл с параметром for.
1.1.14.Цикл с предусловием while, цикл с постусловием do-while.
1.2.7.Понятие функции.
Задание №1 (10 минут)
Ниже записан рекурсивный алгоритм F.

```
void F(int n)
{
  if (n > 0)
  {
    F(n - 1);
    F(n - 1 );
    cout << n;
  }
}
```

Определите, что выведет программа при вызове функции F(4)? Цифры запишите в той последовательности, в которой они выводятся.

(один из вариантов задачи)

Оценка	***Показатели оценки***
5	Расписано пошаговое решение задачи, итоговый результат получен верный.
4	Расписано пошаговое решение задачи, итоговый результат получен не верный.
3	Не продемонстрированно пошаговое решение задачи, итоговый результат написан верный.

Дидактическая единица: 1.4 Подпрограммы, составление библиотек подпрограмм
Занятие(-я):
1.2.7.Понятие функции.
1.2.8.Определение и вызов функций.
1.2.9.Определение и вызов функции.
1.2.10.Локальные и глобальные переменные.
1.2.11.Рекурсивные алгоритмы.
1.2.12.Применение рекурсии.
Задание №1 (5 минут)
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, если глобальная и локальная переменные имеют одинаковые имена? Объясните на примерах.

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

Дидактическая единица: 2.1 Разрабатывать алгоритмы для конкретных задач
Занятие(-я):
1.2.9.Определение и вызов функции.
1.2.12.Применение рекурсии.
Задание №1 (10 минут)
Построить алгоритм решения задачи:

`????????? ???????????? ???? ? ???????? ?????.`

(один из вариантов задачи).

Оценка	*Показатели оценки*
5	Алгоритм построен верно.
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков);
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Дидактическая единица: 2.2 Использовать программы для графического отображения алгоритмов
Занятие(-я):

Задание №1 (5 минут)
Оформить алгоритм в он-лайн графическом редакторе (например: programforyou.ru/block-diagram-redactor).

Оценка	*Показатели оценки*
5	Алгоритм построен верно.
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков);
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

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

Занятие(-я):

1.2.8.Определение и вызов функций.

1.2.9.Определение и вызов функции.

1.2.12.Применение рекурсии.

Задание №1 (15 минут)

Написать программу для решения задачи (с использованием рекурсивной функции):

```
????????? ??????????? ???? ? ???????? ?????.
```

(один из вариантов задачи).

Оценка	***Показатели оценки***
5	Задача решена с применением рекурсивной функции, работает верно.
4	Задача решена с применением рекурсивной функции, работает не корректно.
3	Задача решена без применения рекурсивной функции, работает верно.

2.6 Текущий контроль (ТК) № 6 (45 минут)

Тема занятия: 1.2.22.Создание структур.

Метод и форма контроля: Практическая работа (Опрос)

Вид контроля: Отчет

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

Занятие(-я):

1.1.26.Одномерные массивы. Решение задач.

1.1.27.Способы обработки элементов массива.

1.2.1.Двумерные массивы (матрицы).

1.2.5.Двумерные массивы. Решение задач.

1.2.6.Двумерные массивы. Решение задач.

1.2.10.Локальные и глобальные переменные.

1.2.11.Рекурсивные алгоритмы.

1.2.12.Применение рекурсии.

1.2.17.Основные сведения о структурах. Структуры и функции.

Задание №1 (10 минут)

Ответить на вопросы:

1. Соотнесите термины и родовые понятия

алфавит		множество
идентификатор		набор символов
константа		величина
ключевые слова		единица языка

2. Соотнесите термины и видовые отличия

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

3. Определите, к какому термину относится видовое отличие: «Определяет допустимые значения переменной»

1. тип данных;
2. множество;
3. алфавит;
4. ключевые слова.

4. Конечное множество символов, с помощью которых записывается текст – это ...

1. алфавит;
2. ключевые слова;
3. тип данных;
4. константа.

5. Выберите видовые отличия термина «Ключевые слова»

1. имеют специальное значение для компилятора;
2. нельзя использовать в качестве имени константы, переменной, процедуры, функции и т.п.;

3. являются опорными фрагментами текста;
4. направлены на понимание основной идеи и содержания текста.

Оценка	*Показатели оценки*
5	Отвечено верно на 5 вопросов;
4	Отвечено верно на 4 вопроса;
3	Отвечено верно на 3 вопроса.

Дидактическая единица: 2.4 Работать в среде программирования
Занятие(-я):
1.1.27.Способы обработки элементов массива.
1.2.14.Операторы работы с файлами.
1.2.15.Работа с файлами.
1.2.16.Работа с файлами.
1.2.18.Создание структур.
1.2.21.Структуры со ссылками на себя.
Задание №1 (20 минут)
В среде разработки Visual Studio, создайте структуры для хранения информации об участниках соревнований (Имя, Фамилия, возраст, рост). Организуйте ввод данных о 8 участниках с клавиатуры. Участники распределяются по двум командам (команда 1 и команда 2). Выведите на экран количество участников в команде 1, далее в отдельных строках фамилии участников, в отдельной строке количество участников команды 2, далее в отдельных строках фамилии участников.

(*представлен один из вариантов задач)*

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

Дидактическая единица: 2.6 Оформлять код программы в соответствии со стандартом кодирования
Занятие(-я):
1.2.6.Двумерные массивы. Решение задач.
1.2.13.Решение задач с применением рекрусивных функций.
1.2.14.Операторы работы с файлами.
1.2.15.Работа с файлами.
1.2.16.Работа с файлами.

1.2.18.Создание структур.
1.2.19.Массивы структур.
1.2.20.Указатели на структуры.
1.2.21.Структуры со ссылками на себя.

Задание №1 (5 минут)

Оформите код написанной программ в соответствии со стандартом кодирования.

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

Дидактическая единица: 2.7 Выполнять проверку, отладку кода программы

Занятие(-я):

1.2.8.Определение и вызов функций.
1.2.9.Определение и вызов функции.
1.2.13.Решение задач с применением рекрусивных функций.

Задание №1 (10 минут)

Выполните отладку программы и пошаговое выполнение. Результаты пошагового выполнения представьте в виде скриншотов.

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

2.7 Текущий контроль (ТК) № 7 (40 минут)

Тема занятия: 2.1.7.Работа с классами. Решение задач.

Метод и форма контроля: Практическая работа (Опрос)

Вид контроля: Практическая работа с применением ИКТ

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):

2.1.1.Поточный ввод-вывод информации в языке C++.

2.1.2.Основы объектно-ориентированного программирования.
2.1.3.Класс. Объект класса. Наследование.
2.1.4.Создание собственных классов. Установление отношений дружественности и наследования между классами.

Задание №1 (10 минут)

Дайте ответы на вопросы теста:

1. Что означает аббревиатура ООП:

1. объектный образ в программировании;
2. объектно-ориентированное программирование;
3. объективно ориентированное программирование.

2. Принцип инкапсуляции обеспечивает:

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

3. Укажите правильный вариант определения класса в программе:

1. class Test {
public: int a;
};
2. class Test {
private: float x;
}
3. оба варианта правильные.

4. Спецификатор доступа private обеспечивает:

1. доступность членов класса в методах данного класса и в дружественных функциях данного класса;
2. доступность членов класса в методах данного класса, в дружественных функциях данного класса и в методах наследников данного класса;
3. доступность членов класса во всех функциях программы.

5. Спецификатор доступа protected обеспечивает:

1. доступность членов класса в методах данного класса и в дружественных функциях данного класса;
2. доступность членов класса в методах данного класса, в дружественных функциях данного класса и в методах наследников данного класса;
3. доступность членов класса во всех функциях программы.

6. Спецификатор доступа public обеспечивает:

1. доступность членов класса в методах данного класса и в дружественных функциях данного класса;
2. доступность членов класса в методах данного класса, в дружественных функциях данного класса и в методах наследников данного класса;
3. 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);
...
```

1. a = 45;
2. программа не запустится, так как доступ к полю «a» необходимо получить, используя операцию: L->a = 45;
3. программа не запустится, так как переменная «a» является закрытой.

8. Конструктор — это:

1. специальный метод класса с тем же именем, что и сам класс;
2. специальный метод класса, не имеющий параметров и не возвращающий никакого значения;
3. механизм создания новых объектов класса.

9. В классе может быть только:

1. единственный конструктор;
2. два конструктора — по умолчанию и с параметрами;
3. произвольное количество конструкторов.

10. Деструктор — это:

1. специальный метод класса с тем же именем, что и сам класс с префиксом — тильдой (~);
2. автоматически создаваемый метод класса, предназначенный для удаления объектов класса;
3. механизм разрушения объектов класса.

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

Дидактическая единица: 1.4 Подпрограммы, составление библиотек подпрограмм
Занятие(-я):
1.2.13.Решение задач с применением рекрусивных функций.
1.2.14.Операторы работы с файлами.
1.2.19.Массивы структур.
1.2.20.Указатели на структуры.
1.2.21.Структуры со ссылками на себя.
1.2.22.Создание структур.
Задание №1 (10 минут)
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, если глобальная и локальная переменные имеют одинаковые имена? Объясните на примерах.

Оценка	*Показатели оценки*
5	3 правильных ответа;
4	2 правильных ответа;
3	1 правильный ответ;

Дидактическая единица: 2.6 Оформлять код программы в соответствии со стандартом кодирования
Занятие(-я):
1.2.22.Создание структур.
2.1.1.Поточный ввод-вывод информации в языке C++.
2.1.2.Основы объектно-ориентированного программирования.
2.1.4.Создание собственных классов. Установление отношений дружественности и наследования между классами.
2.1.5.Решение задач с использованием классов.
2.1.6.Работа с классами. Решение задач.
Задание №1 (5 минут)
Оформить код программы в соответствии со стандартом кодирования

Оценка	*Показатели оценки*
3	Созданы классы, 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.

Дидактическая единица: 2.4 Работать в среде программирования
Занятие(-я):
1.2.22.Создание структур.
Задание №1 (10 минут)
Написать программу для следующей задачи:

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.

(один из вариантов задачи)

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

Дидактическая единица: 2.7 Выполнять проверку, отладку кода программы
Занятие(-я):
2.1.4.Создание собственных классов. Установление отношений дружественности и наследования между классами.
2.1.5.Решение задач с использованием классов.
2.1.6.Работа с классами. Решение задач.
Задание №1 (5 минут)
Выполнить проверку и отладку кода программы

Оценка	***Показатели оценки***
5	Логика программы верна, работает корректно
4	Логика программы верна, работает корректно
3	Программа работает не корректно.

2.8 Текущий контроль (ТК) № 8 (45 минут)

Тема занятия: 2.1.10.Решение задач со строками.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Отчет
Дидактическая единица: 2.4 Работать в среде программирования
Занятие(-я):
2.1.8.Класс string, использование, инициализация строк.
2.1.9.Работа со строками, используя метод string.
Задание №1 (35 минут)
Написать программу в среде программирования VisualStudio:

Макс писал сочинение по литературе. Сочинение длинное, мыслей много, а писать от руки Макс не любит. Поэтому он стал сокращать слова, оставляя только первую и последнюю буквы с дефисом между ними.
Дано слово. Выведи, как его написал Макс.
Пример ввода1:

```
sekunda
```

Пример выхода1:

```
s-a
```

Пример ввода2:

```
key
```

ример выхода2:

```
k-y
```

(Представлен один из вариантов)

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

Дидактическая единица: 2.6 Оформлять код программы в соответствии со стандартом кодирования
Занятие(-я):
2.1.7.Работа с классами. Решение задач.
2.1.8.Класс string, использование, инициализация строк.
2.1.9.Работа со строками, используя метод string.
Задание №1 (5 минут)

Оформите код написанных программ в соответствии со стандартом кодирования.

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

Дидактическая единица: 2.7 Выполнять проверку, отладку кода программы
Занятие(-я):
2.1.7.Работа с классами. Решение задач.
2.1.9.Работа со строками, используя метод string.
Задание №1 (5 минут)
Выполните проверку и отладку программы

Оценка	*Показатели оценки*
5	Программа скомпилирована, отлажена, проект собран, создан .EXE файл
4	Программа скомпилирована, отлажена
3	Программа скомпилирована, работает с ошибками

2.9 Текущий контроль (ТК) № 9 (45 минут)

Тема занятия: 2.1.26.Создание Windows Forms. Разработка игрового приложения (разработка алгоритма работы программы).
Метод и форма контроля: Проект (Информационно-аналитический)
Вид контроля: Отчет
Дидактическая единица: 1.1 Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
Занятие(-я):
1.2.13.Решение задач с применением рекрусивных функций.
Задание №1 (5 минут)
Составьте алгоритм работы проекта "Замена букв".

Оценка	*Показатели оценки*
5	Алгоритм отображает работу программы, функций.
4	Алгоритм отображает работу программы, работа функций не отображена. Составлен словесный алгоритм

3	Словесный алгоритм частично отражает работу программы, функций

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

Задание №1 (10 минут)
Составьте алгоритм работы проекта "Замена букв".

Оценка	*Показатели оценки*
5	Алгоритм отображает работу программы, функций.
4	Алгоритм отображает работу программы, работа функций не отображена. Составлен словесный алгоритм
3	Словесный алгоритм частично отражает работу программы, функций

Дидактическая единица: 2.2 Использовать программы для графического отображения алгоритмов
Занятие(-я):

Задание №1 (10 минут)
Оформить схему алгоритма в программе для графического отображения алгоритмов.

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

Дидактическая единица: 2.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования
Занятие(-я):
2.1.14.Визуальные компоненты.
2.1.16.Создание WindowsForms. Калькулятор.
Задание №1 (20 минут)

Написать программу по алгоритму проекта "Замена букв"

Оценка	***Показатели оценки***
5	Программа написана самостоятельно, все функции реализованы
4	Программа написана самостоятельно, функции реализованы частично
3	Программа написана с помощью преподавателя, функции реализованы частично

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
Текущий контроль №8
Текущий контроль №9

Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: По выбору выполнить 1 теоретическое задание и 3 практических задания
Дидактическая единица для контроля:
1.1 Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции

Задание №1
Ответить на тестовые вопросы, в личном кабинете ИАС "ИркАТ" – " ОП.04 Основы алгоритмизации и программирования. Экзамен".

Оценка	*Показатели оценки*
5	Даны верные ответы на 95% вопросов.
4	Даны верные ответы на 75% вопросов.
3	Даны верные ответы на 50% вопросов.

Задание №2
Вопрос 1:
Для решения любой задачи с помощью компьютера необходимо выполнить следующие этапы:
Установите правильную последовательность этапов.

Ответ

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

Вопрос 2:

Соотнесите свойства алгоритма с их описанием:

Ответ

Результативность	алгоритм должен обеспечить выдачу результата решения задачи на печать, на экран монитора или в файл
Конечность	алгоритм должен приводить к решению задачи обязательно за конечное время
Эффективность	позволяет решить задачу за приемлемое для разработчика время
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	неоднозначность толкования алгоритма недопустима

Вопрос 3:

Выберите тип алгоритма, описанного ниже:

Набор команд (указаний), выполняемых последовательно друг за другом

Ответ

- 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
Ответ

- Линейный
- Разветвляющийся
- Циклический

Вопрос 5:
Выберите тип алгоритма, описанного ниже:
Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными
Ответ

- Линейный
- Разветвляющийся
- Циклический

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Даны верные ответы на 1 вопрос.

Задание №3
Вопрос 1:
Для решения любой задачи с помощью компьютера необходимо выполнить следующие этапы:
Установите правильную последовательность этапов.
Ответ

1	Постановка задачи
2	Математическое моделирование

3	Алгоритмизация задач
4	Программирование
5	Анализ результатов

Вопрос 2:
Соотнесите свойства алгоритма с их описанием:
Ответ

Результативность	алгоритм должен обеспечить выдачу результата решения задачи на печать, на экран монитора или в файл
Конечность	алгоритм должен приводить к решению задачи обязательно за конечное время
Эффективность	позволяет решить задачу за приемлемое для разработчика время
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	неоднозначность толкования алгоритма недопустима

Вопрос 3:
Выберите тип алгоритма, описанного ниже:
Набор команд (указаний), выполняемых последовательно друг за другом
Ответ

- 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
Ответ

- Линейный
- Разветвляющийся
- Циклический

Вопрос 5:
Выберите тип алгоритма, описанного ниже:
Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными
Ответ

- Линейный
- Разветвляющийся

1. Циклический

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №4
Вопрос 1:
Для решения любой задачи с помощью компьютера необходимо выполнить следующие этапы:
Установите правильную последовательность этапов.
Ответ

1	Постановка задачи
2	Математическое моделирование
3	Алгоритмизация задач

4	Программирование
5	Анализ результатов

Вопрос 2:
Соотнесите свойства алгоритма с их описанием:
Ответ

Результативность	алгоритм должен обеспечить выдачу результата решения задачи на печать, на экран монитора или в файл
Конечность	алгоритм должен приводить к решению задачи обязательно за конечное время
Эффективность	позволяет решить задачу за приемлемое для разработчика время
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	неоднозначность толкования алгоритма недопустима

Вопрос 3:
Выберите тип алгоритма, описанного ниже:
Набор команд (указаний), выполняемых последовательно друг за другом
Ответ

- 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
Ответ

- Линейный
- Разветвляющийся
- Циклический

Вопрос 5:
Выберите тип алгоритма, описанного ниже:
Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными
Ответ

- Линейный
- Разветвляющийся
- Циклический

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №5
Вопрос 1:
Для решения любой задачи с помощью компьютера необходимо выполнить следующие этапы:
Установите правильную последовательность этапов.
Ответ

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

Вопрос 2:
Соотнесите свойства алгоритма с их описанием:
Ответ

Результативность	алгоритм должен обеспечить выдачу результата решения задачи на печать, на экран монитора или в файл
Конечность	алгоритм должен приводить к решению задачи обязательно за конечное время
Эффективность	позволяет решить задачу за приемлемое для разработчика время
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	неоднозначность толкования алгоритма недопустима

Вопрос 3:
Выберите тип алгоритма, описанного ниже:
Набор команд (указаний), выполняемых последовательно друг за другом
Ответ

- 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
Ответ

- Линейный
- Разветвляющийся
- Циклический

Вопрос 5:
Выберите тип алгоритма, описанного ниже:
Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными
Ответ

- Линейный
- Разветвляющийся
- Циклический

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №6
Вопрос 1:
Для решения любой задачи с помощью компьютера необходимо выполнить следующие этапы:
Установите правильную последовательность этапов.
Ответ

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

Вопрос 2:

Соотнесите свойства алгоритма с их описанием:
Ответ

Результативность	алгоритм должен обеспечить выдачу результата решения задачи на печать, на экран монитора или в файл
Конечность	алгоритм должен приводить к решению задачи обязательно за конечное время
Эффективность	позволяет решить задачу за приемлемое для разработчика время
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	неоднозначность толкования алгоритма недопустима

Вопрос 3:
Выберите тип алгоритма, описанного ниже:
Набор команд (указаний), выполняемых последовательно друг за другом
Ответ

- 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

Ответ

- Линейный
- Разветвляющийся
- Циклический

Вопрос 5:
Выберите тип алгоритма, описанного ниже:
Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными
Ответ

- Линейный
- Разветвляющийся
- Циклический

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №7
Вопрос 1:
Для решения любой задачи с помощью компьютера необходимо выполнить следующие этапы:
Установите правильную последовательность этапов.
Ответ

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

Вопрос 2:
Соотнесите свойства алгоритма с их описанием:
Ответ

Результативность	алгоритм должен обеспечить выдачу результата решения задачи на печать, на экран монитора или в файл
Конечность	алгоритм должен приводить к решению задачи обязательно за конечное время
Эффективность	позволяет решить задачу за приемлемое для разработчика время
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	неоднозначность толкования алгоритма недопустима

Вопрос 3:
Выберите тип алгоритма, описанного ниже:
Набор команд (указаний), выполняемых последовательно друг за другом
Ответ

- 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
Ответ

- Линейный
- Разветвляющийся
- Циклический

Вопрос 5:
Выберите тип алгоритма, описанного ниже:
Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными
Ответ

- Линейный
- Разветвляющийся
- Циклический

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №8
Ответить на вопросы:
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

1. линейный;
2. разветвляющийся;
3. 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

1. линейный;
2. разветвляющийся;
3. циклический.

Вопрос 5 Выберите тип алгоритма, описанного ниже:
Алгоритм, предусматривающий многократное повторение одного и того же

действия над новыми исходными данными

1. линейный;
2. разветвляющийся;
3. 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 №9
Ответить на вопросы:
1. Объясните на примере принцип работы полной и краткой формы условного оператора.
2. как показать условный оператор на схеме алгоритма? Приведите примеры.
3. Можно ли использовать условный оператор внутри другого условного оператора (например, в ветви else)?

Оценка	*Показатели оценки*
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Задание №10
Ответить на вопросы:
1. Из чего состоит итерация для оператора while? Объясните на примере.
2. Объясните механизм работы опреатора do-while на примерах.
3. Запишите в общем виде оператор цикла for. Какие ключевые слова пи этом используются.

Оценка	*Показатели оценки*
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Дидактическая единица для контроля:

1.2 Эволюцию языков программирования, их классификацию, понятие системы программирования

Задание №1

Ответить на тестовые вопросы, в личном кабинете ИАС "ИркАТ" – " ОП.04 Основы алгоритмизации и программирования. Экзамен".

Оценка	***Показатели оценки***
5	Даны верные ответы на 95% вопросов.
4	Даны верные ответы на 75% вопросов.
3	Даны верные ответы на 50% вопросов.

Задание №2

Ответить на вопросы:

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:

1. C++;
2. Basic;
3. Java.

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

1. Commonlisp.
2. Planner.

3. Occam.
4. C++.
5. Java.

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

1. Commonlisp.
2. Planner.
3. Occam.
4. C++.
5. 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

Задание №3

Вопрос 1:

Установите правильную хронологию создания языков программирования:

Ответ

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

Вопрос 2:

Соотнесите годы создания языков программирования

Ответ

C#	2000
Бейсик (Basic)	1963
Паскаль (Pascal)	1971
C++	1986

Вопрос 3:

Выберите процедурный язык программирования:

Ответ

- C++
- Basic
- Java

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

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

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

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

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса
4	Даны верные ответы на 2 вопроса
3	Даны верные ответы на 1 вопрос

Задание №4
Вопрос 1:
Установите правильную хронологию создания языков программирования:
Ответ

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

Вопрос 2:
Соотнесите годы создания языков программирования
Ответ

C#	2000
Бейсик (Basic)	1963
Паскаль (Pascal)	1971
C++	1986

Вопрос 3:
Выберите процедурный язык программирования:
Ответ

- C++
- Basic
- Java

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

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

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

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

Оценка	*Показатели оценки*
5	Даны верные ответы на 3 вопроса
4	Даны верные ответы на 2 вопроса
3	Даны верные ответы на 1 вопрос

Задание №5

Вопрос 1:

Установите правильную хронологию создания языков программирования:

Ответ

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

Вопрос 2:

Соотнесите годы создания языков программирования

Ответ

C#	2000
Бейсик (Basic)	1963
Паскаль (Pascal)	1971
C++	1986

Вопрос 3:

Выберите процедурный язык программирования:

Ответ

- C++
- Basic
- Java

Вопрос 4:

Какие языки программирования предназначены для решения задач искусственного

интеллекта:
Ответ

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

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

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

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса
4	Даны верные ответы на 2 вопроса
3	Дан верный ответ на 1 вопрос.

Задание №6
Вопрос 1:
Установите правильную хронологию создания языков программирования:
Ответ

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

Вопрос 2:
Соотнесите годы создания языков программирования
Ответ

C#	2000
Бейсик (Basic)	1963
Паскаль (Pascal)	1971
C++	1986

Вопрос 3:
Выберите процедурный язык программирования:
Ответ

- C++
- Basic
- Java

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

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

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

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

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса

4	Даны верные ответы на 2 вопроса
3	Дан верный ответ на 1 вопрос.

Задание №7

Вопрос 1:

Установите правильную хронологию создания языков программирования:

Ответ

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

Вопрос 2:

Соотнесите годы создания языков программирования

Ответ

C#	2000
Бейсик (Basic)	1963
Паскаль (Pascal)	1971
C++	1986

Вопрос 3:

Выберите процедурный язык программирования:

Ответ

- C++
- Basic
- Java

Вопрос 4:

Какие языки программирования предназначены для решения задач искусственного интеллекта:

Ответ

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

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

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

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса
4	Даны верные ответы на 2 вопроса
3	Дан верный ответ на 1 вопрос.

Задание №8
Вопрос 1:
Установите правильную хронологию создания языков программирования:
Ответ

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

Вопрос 2:
Соотнесите годы создания языков программирования
Ответ

C#	2000
Бейсик (Basic)	1963
Паскаль (Pascal)	1971
C++	1986

Вопрос 3:
Выберите процедурный язык программирования:
Ответ

- C++
- Basic
- Java

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

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

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

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

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса
4	Даны верные ответы на 2 вопроса
3	Дан верный ответ на 1 вопрос.

Задание №9
Вопрос 1:
Установите правильную хронологию создания языков программирования:
Ответ

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

4	C#

Вопрос 2:
Соотнесите годы создания языков программирования
Ответ

C#	2000
Бейсик (Basic)	1963
Паскаль (Pascal)	1971
C++	1986

Вопрос 3:
Выберите процедурный язык программирования:
Ответ

- C++
- Basic
- Java

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

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

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

- Commonlisp
- Planner
- Occam
- C++

- Java

Оценка	*Показатели оценки*
5	Даны верные ответы на 3 вопроса
4	Даны верные ответы на 2 вопроса
3	Дан верный ответ на 1 вопрос.

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

Задание №1

Ответить на тестовые вопросы, в личном кабинете ИАС "ИркАТ" – " ОП.04 Основы алгоритмизации и программирования. Экзамен".

Оценка	*Показатели оценки*
5	Даны верные ответы на 95% вопросов.
4	Даны верные ответы на 75% вопросов.
3	Даны верные ответы на 50% вопросов.

Задание №2

Ответить на вопросы:

Вопрос 1. Выберите видовые отличия термина "Язык программирования"

1. предназначен для записи программ;
2. имеет определенный синтаксис;
3. предназначен для передачи информации;
4. машиноориентированный, низкого уровня.

Вопрос 2 Выберите видовые отличия термина "Алгоритм"

1. ориентирована на пользователя;
2. выполняется за конечное число шагов;
3. предназначен для записи программ;
4. имеет определенный синтаксис.

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

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

Вопрос 4 Выберите родовое понятие для термина "Моделирование"

1. система познания;
2. набор объектов;
3. 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}:

1. if (C*D>E+F) Max=C*D; else Max:=E+F;
2. X=C*D; Y=E+F; if (X>Y) Max=X; else MAX=Y;

3. оба варианта правильные.

Вопрос 8 Выберите пример правильного идентификатора в языке C++

1. Fr_5;
2. 10Sd;
3. 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;
}
...
```

1. S=-7; N=1; P=0;
2. S=6; N=2; P=10;
3. 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;
}
...
```

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

Вопрос 11 Укажите правильный вариант определения класса в программе:
1. class Test
{
public:
int a;
};
2. class Test
{
private:
float x;
}
3. 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

Задание №3
Ответить на вопросы:
1. Что такое зарезервированные (ключевые слова)? Приведите примеры ключевых слов.
2. Верно ли, что в любой программе на C++ обязательно должны использоваться

переменные и константы? Почему?
3. Чем отличаются операции "/" и "%"? Приведите примеры.

Оценка	***Показатели оценки***
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Задание №4
Вопрос 1:
Выберите правильный вариант использования условного оператора if для нахождения MAX{C*D, E+F}:
Ответ

- f (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++
Ответ

- 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;
}
...
```
Ответ

- 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;
}
...
```
Ответ

- 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:
Дан массив A=(-9 0 2 -6 5 4). Что будет сформировано в переменной X после выполнения следующего фрагмента программы?

```
...
X=0;
for (i=0; i<6; i++)
if ((A[i]<0) && (A[i] %2 != 0))
X=X+A[i];
...
```
Ответ

- сумма отрицательных нечетных элементов целочисленного массива A (X = -9)
- произведение положительных четных элементов целочисленного массива A (X = 8)
- количество ненулевых элементов массива A (X = 5)

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Даны верные ответы на 1 вопрос.

Задание №5
Вопрос 1:
Выберите правильный вариант использования условного оператора if для нахождения MAX{C*D, E+F}:
Ответ

- f (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++
Ответ

- 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;
}
...
```
Ответ

- 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;
}
...
```
Ответ

- 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:
Дан массив A=(-9 0 2 -6 5 4). Что будет сформировано в переменной X после выполнения следующего фрагмента программы?

```
...
X=0;
for (i=0; i<6; i++)
if ((A[i]<0) && (A[i] %2 != 0))
X=X+A[i];
...
```

Ответ

- сумма отрицательных нечетных элементов целочисленного массива A (X = -9)
- произведение положительных четных элементов целочисленного массива A (X = 8)
- количество ненулевых элементов массива A (X = 5)

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верные ответы на 1 вопрос.

Задание №6
Вопрос 1:
Выберите правильный вариант использования условного оператора if для нахождения MAX{C*D, E+F}:
Ответ

- f (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++
Ответ

- 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;
}
...
```

Ответ

- 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;
}
...
```

Ответ

- 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:
Дан массив A=(-9 0 2 -6 5 4). Что будет сформировано в переменной X после выполнения следующего фрагмента программы?

```
...
X=0;
for (i=0; i<6; i++)
```

```
if ((A[i]<0) && (A[i] %2 != 0))
X=X+A[i];
...
```
Ответ

- сумма отрицательных нечетных элементов целочисленного массива A (X = -9)
- произведение положительных четных элементов целочисленного массива A (X = 8)
- количество ненулевых элементов массива A (X = 5)

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №7

Вопрос 1:

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

Ответ

- f (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++

Ответ

- 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;
}
```
...
Ответ

- 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;
}
```
...
Ответ

- 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:
Дан массив A=(-9 0 2 -6 5 4). Что будет сформировано в переменной X после выполнения следующего фрагмента программы?

...
X=0;
for (i=0; i<6; i++)
if ((A[i]<0) && (A[i] %2 != 0))
X=X+A[i];
...
Ответ

- сумма отрицательных нечетных элементов целочисленного массива A (X = -9)
- произведение положительных четных элементов целочисленного массива A (X = 8)
- количество ненулевых элементов массива A (X = 5)

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №8
Вопрос 1:
Выберите правильный вариант использования условного оператора if для нахождения MAX{C*D, E+F}:
Ответ

- f (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++
Ответ

- 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;
}
...
```

Ответ

- 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;
}
...
```

Ответ

- 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:
Дан массив A=(-9 0 2 -6 5 4). Что будет сформировано в переменной X после выполнения следующего фрагмента программы?
...
X=0;
for (i=0; i<6; i++)
if ((A[i]<0) && (A[i] %2 != 0))
X=X+A[i];
...
Ответ

- сумма отрицательных нечетных элементов целочисленного массива A (X = -9)
- произведение положительных четных элементов целочисленного массива A (X = 8)
- количество ненулевых элементов массива A (X = 5)

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №9
Вопрос 1:
Выберите правильный вариант использования условного оператора if для нахождения MAX{C*D, E+F}:
Ответ

- f (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++
Ответ

- 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;
}
```
...
Ответ

- 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;
}
```
...
Ответ

- 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:
Дан массив A=(-9 0 2 -6 5 4). Что будет сформировано в переменной X после выполнения следующего фрагмента программы?
...
X=0;
for (i=0; i<6; i++)
if ((A[i]<0) && (A[i] %2 != 0))
X=X+A[i];
...
Ответ

- сумма отрицательных нечетных элементов целочисленного массива A (X = -9)
- произведение положительных четных элементов целочисленного массива A (X = 8)
- количество ненулевых элементов массива A (X = 5)

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №10
Вопрос 1:
Выберите правильный вариант использования условного оператора if для нахождения MAX{C*D, E+F}:
Ответ

- f (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++
Ответ

- 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;
}
...
```

Ответ

- 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;
}
...
```

Ответ

- 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:
Дан массив A=(-9 0 2 -6 5 4). Что будет сформировано в переменной X после выполнения следующего фрагмента программы?

```
...
X=0;
for (i=0; i<6; i++)
if ((A[i]<0) && (A[i] %2 != 0))
X=X+A[i];
...
```

Ответ

- сумма отрицательных нечетных элементов целочисленного массива A (X = -9)
- произведение положительных четных элементов целочисленного массива A (X = 8)
- количество ненулевых элементов массива A (X = 5)

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №11
Ответить на вопросы:
1. Данные каких типов могут быть элемнтами массива? Приведите примеры.
2. Как получить доступ к конкретному элементу массива? Приведите примеры.
3. Можно ли в программе объявить пользовательский тип "массива"? Как это сделать? Приведите примеры.

Оценка	***Показатели оценки***
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Дидактическая единица для контроля:

1.4 Подпрограммы, составление библиотек подпрограмм

Задание №1

Ответить на тестовые вопросы, в личном кабинете ИАС "ИркАТ" – " ОП.04 Основы алгоритмизации и программирования. Экзамен".

Оценка	***Показатели оценки***
5	Даны верные ответы на 95% вопросов.
4	Даны верные ответы на 75% вопросов.
3	Даны верные ответы на 50% вопросов.

Задание №2

Ответить на вопросы:

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

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

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

1. разницы нет;
2. глобальные переменные не могут использоваться в подпрограммах, для этого служат локальные переменные;
3. 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:

1. int Area(int A, int B)
{ float S;
Area=A*B;}
2. int Area(int A, int B)
{ int S;
S=A*B; return S;}

3. 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;
}
```

1. 10 10;
2. 5 10;
3. 5 5.

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

1. да;
2. нет;
3. 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 №3

Ответить на вопросы:

1. Что такое возвращаемое значение в функции? Приведите примеры.
2. Напишите структуру функции в общем виде и объясните, для чего используется

каждый раздел функции. Приведите примеры определения функции.
3. Что такое формальные и фактические параметры в функции? Приведите примеры.

Оценка	***Показатели оценки***
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Задание №4
Ответить на вопросы:
1. Что такое "структура" в программировании с точки зрения типов данных?
2. Какие требования предъявляются к именам полей структур? Поясните на примерах.
3. Как правильно присваивать значения полям структуры? Поясните на примерах.

Оценка	***Показатели оценки***
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Задание №5
Вопрос 1:
В чем разница между формальными и фактическими параметрами?
Ответ

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

Вопрос 2:
В чем разница между глобальными и локальными переменными?
Ответ

- 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:
Ответ

- 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;
}
```

Ответ

- 10 10
- 5 10
- 5 5

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

- да
- нет
- должны соответствовать в прототипе и определении

Оценка	*Показатели оценки*
5	Дааны верные ответы на 3 вопроса.
4	Дааны верные ответы на 2 вопроса.
3	Дааны верные ответы на 1 вопрос.

Задание №6

Вопрос 1:

В чем разница между формальными и фактическими параметрами?

Ответ

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

Вопрос 2:

В чем разница между глобальными и локальными переменными?

Ответ

- 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:

Ответ

- 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;
}
```
Ответ

- 10 10
- 5 10
- 5 5

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

- да
- нет
- должны соответствовать в прототипе и определении

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса
4	Даны верные ответы на 2 вопроса
3	Дан верный ответ на 1 вопрос

Задание №7
Вопрос 1:
В чем разница между формальными и фактическими параметрами?
Ответ

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

- формальные параметры используются при описании подпрограммы, а фактические — при вызове подпрограммы;

Вопрос 2:
В чем разница между глобальными и локальными переменными?
Ответ

- 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:
Ответ

- 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;
}
```

Ответ

- 10 10
- 5 10
- 5 5

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

- да
- нет
- должны соответствовать в прототипе и определении

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №8
Вопрос 1:
В чем разница между формальными и фактическими параметрами?
Ответ

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

Вопрос 2:
В чем разница между глобальными и локальными переменными?
Ответ

- 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:

Ответ

- 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;
}
```

Ответ

- 10 10
- 5 10
- 5 5

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

- да
- нет
- должны соответствовать в прототипе и определении

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №9

Вопрос 1:

В чем разница между формальными и фактическими параметрами?

Ответ

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

Вопрос 2:

В чем разница между глобальными и локальными переменными?

Ответ

- 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:

Ответ

- 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;
}
```
Ответ

- 10 10
- 5 10
- 5 5

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

- да
- нет
- должны соответствовать в прототипе и определении

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №10
Вопрос 1:
В чем разница между формальными и фактическими параметрами?
Ответ

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

Вопрос 2:
В чем разница между глобальными и локальными переменными?
Ответ

- 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:
Ответ

- 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;
}
```

Ответ

- 10 10
- 5 10
- 5 5

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

- да

- нет
- должны соответствовать в прототипе и определении

Оценка	*Показатели оценки*
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №11

Вопрос 1:

В чем разница между формальными и фактическими параметрами?

Ответ

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

Вопрос 2:

В чем разница между глобальными и локальными переменными?

Ответ

- 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:

Ответ

- 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;
}
```

Ответ

- 10 10
- 5 10
- 5 5

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

- да
- нет
- должны соответствовать в прототипе и определении

Оценка	***Показатели оценки***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

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

Ответить на тестовые вопросы, в личном кабинете ИАС "ИркАТ" – " ОП.04 Основы алгоритмизации и программирования. Экзамен".

Оценка	***Показатели оценки***
5	Даны верные ответы на 95% вопросов.
4	Даны верные ответы на 75% вопросов.
3	Даны верные ответы на 50% вопросов.

Задание №2
Ответить на вопросы:
Вопрос 1. Принцип инкапсуляции обеспечивает:

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

Вопрос 2. Укажите правильный вариант инициализации целочисленных констант A и B в классе:
1. klass::klass(int a, int b) : A(a), B(b)
{
• • •
}
2. klass::klass(int a, int b) : A= a, B = b
{
• • •
}
3. klass::klass(int a, int b) : A(a) B(b)
{
• • •
}

Вопрос 3. Наследование — это:

1. возможность использования базовых библиотек языка C++ в своих программах;
2. условия, описывающие последовательность вызова конструкторов для объектов классов, используемых в программе;

3. механизм создания производных классов, на базе уже имеющихся.

Вопрос 4. Укажите верную последовательность выполнения деструкторов:

1. сначала выполняются операторы деструктора базового класса, затем операторы деструктора порожденного класса;
2. сначала выполняются операторы деструктора порожденного класса, затем операторы деструктора базового класса;
3. операторы деструктора порожденного класса выполняются одновременно с операторами деструктора базового класса.

Вопрос 5. Полиморфизм — это:

1. возможность программного кода работать с разными объектами одинаковым образом;
2. возможность изменения программного кода в зависимости от решаемых задач;
3. 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

Задание №3
Вопрос 1:
Принцип инкапсуляции обеспечивает:
Ответ

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

Вопрос 2:
Что означает аббревиатура ООП?
Ответ

- объектный образ в программировании
- объектно-ориентированное программирование
- объективно ориентированное программирование

Вопрос 3:
Наследование — это:
Ответ

- 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:
Ответ

1	выполняет действия, описанные программистом
2	присваевает указателям пустые значения
3	выполняет действия по разрушению объекта

Вопрос 5:
Полиморфизм — это
Ответ

- 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*
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Даны верные ответы на 1 вопрос.

Задание №4

Вопрос 1:

Принцип инкапсуляции обеспечивает:

Ответ

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

Вопрос 2:

Что означает аббревиатура ООП?

Ответ

- объектный образ в программировании
- объектно-ориентированное программирование
- объективно ориентированное программирование

Вопрос 3:

Наследование — это:

Ответ

- 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:

Ответ

1 выполняет действия, описанные программистом

2	присваевает указателям пустые значения
3	выполняет действия по разрушению объекта

Вопрос 5:
Полиморфизм — это
Ответ

- 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***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №5
Вопрос 1:
Принцип инкапсуляции обеспечивает:
Ответ

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

Вопрос 2:
Что означает аббревиатура ООП?
Ответ

- объектный образ в программировании
- объектно-ориентированное программирование
- объективно ориентированное программирование

Вопрос 3:
Наследование — это:
Ответ

- 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:
Ответ

1	выполняет действия, описанные программистом
2	присваевает указателям пустые значения
3	выполняет действия по разрушению объекта

Вопрос 5:
Полиморфизм — это
Ответ

- 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***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Даны верные ответы на 1 вопрос.

Задание №6
Вопрос 1:
Принцип инкапсуляции обеспечивает:

Ответ

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

Вопрос 2:
Что означает аббревиатура ООП?
Ответ

- объектный образ в программировании
- объектно-ориентированное программирование
- объективно ориентированное программирование

Вопрос 3:
Наследование — это:
Ответ

- 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:
Ответ

1	выполняет действия, описанные программистом
2	присваевает указателям пустые значения
3	выполняет действия по разрушению объекта

Вопрос 5:
Полиморфизм — это
Ответ

- 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***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Даны верные ответы на 1 вопрос.

Задание №7
Вопрос 1:
Принцип инкапсуляции обеспечивает:
Ответ

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

Вопрос 2:
Что означает аббревиатура ООП?
Ответ

- объектный образ в программировании
- объектно-ориентированное программирование
- объективно ориентированное программирование

Вопрос 3:
Наследование — это:
Ответ

- 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:
Ответ

1	выполняет действия, описанные программистом
2	присваевает указателям пустые значения
3	выполняет действия по разрушению объекта

Вопрос 5:
Полиморфизм — это
Ответ

- 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***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Даны верные ответы на 1 вопрос.

Задание №8
Вопрос 1:
Принцип инкапсуляции обеспечивает:
Ответ

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

Вопрос 2:
Что означает аббревиатура ООП?
Ответ

- объектный образ в программировании
- объектно-ориентированное программирование
- объективно ориентированное программирование

Вопрос 3:
Наследование — это:
Ответ

- 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:
Ответ

1	выполняет действия, описанные программистом
2	присваевает указателям пустые значения
3	выполняет действия по разрушению объекта

Вопрос 5:
Полиморфизм — это
Ответ

- 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***
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Задание №9

Вопрос 1:

Принцип инкапсуляции обеспечивает:

Ответ

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

Вопрос 2:

Что означает аббревиатура ООП?

Ответ

- объектный образ в программировании
- объектно-ориентированное программирование
- объективно ориентированное программирование

Вопрос 3:

Наследование — это:

Ответ

- 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:

Ответ

1	выполняет действия, описанные программистом
2	присваевает указателям пустые значения
3	выполняет действия по разрушению объекта

Вопрос 5:

Полиморфизм — это
Ответ

- 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*
5	Даны верные ответы на 3 вопроса.
4	Даны верные ответы на 2 вопроса.
3	Дан верный ответ на 1 вопрос.

Дидактическая единица для контроля:
2.1 Разрабатывать алгоритмы для конкретных задач
Задание №1
Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:
Напишите программу, массив в котором заполняется случайными числами в интервале [0,5], вводится число X и находятся все значения, равные X.
Пример:
Массив:
13 15 13 34 5
Что ищем:
13
Нашли: A[1]=13, A[3]=13
Пример:
Массив:
5 7 9 34 6
Что ищем:
15
Ничего не нашли.

Оценка	*Показатели оценки*
5	Алгоритм построен верно.
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков).

3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Задание №2
Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:
Дано целое число N (> 0). С помощью операций деления нацело и взятия остатка от деления определить, имеется ли в записи числа N цифра «2». Если имеется, то вывести TRUE, если нет — вывести FALSE.

Оценка	***Показатели оценки***
5	Алгоритм построен верно;
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков);
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Задание №3
Разработайте алгоритм для задачи:

Определить, принадлежит ли точка A{x, y) заданной фигуре.

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

Задание №4
Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:
Программа, которая запрашивает стоимость заказа. Если стоимость заказа 1000 и более, то программа должна вывести “Доставка бесплатно.” Иначе программа должна спросить “1. Самовывоз. 2. Доставка курьером.”. Если пользователь вводит 1- программа должна вывести “Выбран самовывоз”, если 2 “Выбрана доставка курьером”. В конце программа должна вывести сумму заказа с учетом доставки. Доставка курьером увеличивает стоимость заказа на 300 руб., самовывоз увеличивает стоимость заказа на 100 руб. Программа должна вывести стоимость заказа (только цифру).

Оценка	*Показатели оценки*
5	Алгоритм построен верно;
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков);
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Задание №5

Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:
Некоторое предприятие ежесуточно расходует X Квт/час электроэнергии – зимой, Y Квт/час – летом и Z квт/час - весной и осенью. Составить программу, вычисляющую расход электроэнергии R для заданного месяца текущего года.

Оценка	*Показатели оценки*
5	Алгоритм построен верно;
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков);
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Задание №6

Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:
Дано натуральное число.
а) Верно ли, что произведение его цифр больше b?
б) Верно ли, что это число k-значное?
Составное условие и вложенный условный оператор не использовать

Оценка	*Показатели оценки*
5	Алгоритм построен верно;
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков);
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Задание №7

Разработайте алгоритм для задачи:

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

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

Задание №8
Разработайте алгоритм для задачи:

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.)

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

Задание №9
Разработайте алгоритм для задачи:

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.

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

Задание №10
Разработайте алгоритм для задачи:

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

Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:

Напишите программу, в котором после заполнения массива случайными числами в интервале [0,100] подсчитывается отдельно среднее значение всех элементов, которые <50, и среднее значение всех элементов, которые больше или равно 50

Пример:

Массив:

4 1 51 4 61

Среднее арифметическое элементов [0,50): 3

Среднее арифметическое элементов [50,100): 56

Оценка	***Показатели оценки***
5	Алгоритм построен верно.
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков).
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Задание №2

Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:

Напишите программу, в котором массив заполняется случайными числами в интервале [-10,10] и отбирающий в другой массив все четные отрицательные числа.

Пример:

Массив А:

-5 6 7 -4 -6 8 -8
Массив B:
-4 -6 -8

Оценка	*Показатели оценки*
5	Алгоритм построен верно.
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков).
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Задание №3
Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:
Программа, получает два целых числа и находит их произведение, не используя операцию умножения. Учитывать, что числа могут быть и отрицательными.
Пример:
Введите два числа:
21 -3
21*(-3)=-63

Оценка	*Показатели оценки*
5	Алгоритм построен верно;
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков);
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Задание №4
Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:
Даны целые положительные числа N и K. Используя только операции сложения и вычитания, найти частное от деления нацело N на K, а также остаток от этого деления.

Оценка	*Показатели оценки*
5	Алгоритм построен верно;

4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков);
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Задание №5

Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:
Определить, является ли заданное шестизначное число счастливым. (Счастливым называют такое шестизначное число, что сумма его первых трех цифр равна сумме его последних трех цифр.)

Оценка	***Показатели оценки***
5	Алгоритм построен верно;
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков);
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Задание №6

Построить алгоритм в виде блок-схемы в онлайн-сервисе (например, programforyou.ru/block-diagram-redactor) для следующей задачи:
Дано натуральное число N. Если оно делится на 4, вывести на экран ответ N=4k (где k – соответствующее частное), если остаток от деления на 4 равен 1 – N=4k+1, если остаток от деления на 4 равен 2 – N=4k+2, если остаток от деления на 4 равен 3 – N=4k+3. Например: 12=4*3, 22=4*5+2.

Оценка	***Показатели оценки***
5	Алгоритм построен верно;
4	Алгоритм построен с небольшими недочетами (например: неправильное соединение блоков);
3	Алгоритм построен с грубыми ошибками (например: неправильно изображены блоки и их действия).

Дидактическая единица для контроля:

2.3 Определять сложность работы алгоритмов

Задание №1

Определить функцию сложности алгоритма по результатам эксперимента:

N	Количество перестановок
5	62

Оценка	*Показатели оценки*
5	Задача решена с ошибками
4	Задача решена, есть замечания по расчетам
3	Задача решена с ошибками

Дидактическая единица для контроля:

2.4 Работать в среде программирования

Задание №1 (из текущего контроля) (20 минут)

В среде разработки Visual Studio напишите программу, которая вычисляет сумму первых n целых положительных четных чисел. Количество суммируемых чисел вводится во время работы программы.

Рекомендуемый вид экрана:

```
?????????? ????? ???????????? ?????? ?????.??????? ????????
?? ?????????? ?????:12????? ?????? 12 ???????????? ??????
????? ????? 156.
```

(*представлен один из вариантов задач)*

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

Задание №2

Написать программу в среде разработки VisualStudio, для задачи:

Напишите программу, заполняющая квадратную матрицу случайными числами в интервале [10,99], и находит максимальный и минимальный элементы в матрице и их индексы.

Пример:

Матрица A:

12 14 67 45

32 87 45 63

69 45 14 11

40 12 35 15

Максимальный элемент A[2,2]=87
Минимальный элемент A[3,4]=11

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

Задание №3
Написать программу в среде разработки VisualStudio, для следующей задачи:
Даны целые положительные числа M, N, число D и набор из M чисел.
Сформировать матрицу размера M x N, у которой первый столбец совпадает с исходным набором чисел, а элементы каждого следующего столбца равны сумме соответствующего элемента предыдущего столбца и числа D (в результате каждая строка матрицы будет содержать элементы арифметической прогрессии).

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

Задание №4
Написать программу в среде разработки VisualStudio, для следующей задачи:
Описать функцию TimeToHMS(T, H, M, S), определяющую по времени T (в секундах) содержащееся в нем количество часов H, минут M и секунд S (T — входной, H, M и S — выходные параметры целого типа).
Используя эту функцию, найти количество часов, минут и секунд для пяти данных отрезков времени T1, T2, . . ., T5.

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

Задание №5
Написать программу в среде разработки VisualStudio, для следующей задачи:
Даны основания и высоты двух равнобедренных трапеций. Найти сумму их периметров и сумму их площадей. (Определить функцию для расчета периметра и

площади равнобедренной трапеции по ее основаниям и высоте.) Площадь трапеции (S)= ((a + b)/2) x h**,** где a и b — длины параллельных сторон, a h — высота трапеции. Периметр трапеции (P) = a + b + 2 √ h 2 + (a - b) 2 /4.

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

Задание №6

Написать программу в среде разработки VisualStudio, для следующей задачи: Написать рекурсивную функцию для вычисления максимального элемента массива из *n* элементов

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

Задание №7

Написать программу в среде разработки VisualStudio, для следующей задачи: Определить комбинированный (структурный) тип для представления информации по горным вершинам, состоящей из названия вершины и ее высоты. Ввести информацию по 5 вершинам. Вывести название самой низкой вершины из всех 5.

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

Задание №8

Написать программу в среде разработки VisualStudio, для следующей задачи: Николаю требуется проверить, возможно ли из представленных отрезков условной длины сформировать треугольник. Для этого он решил создать класс **TriangleChecker**, принимающий только положительные числа. С помощью метода **is_triangle()** возвращаются следующие значения (в зависимости от ситуации):

– **Ура, можно построить треугольник!;**

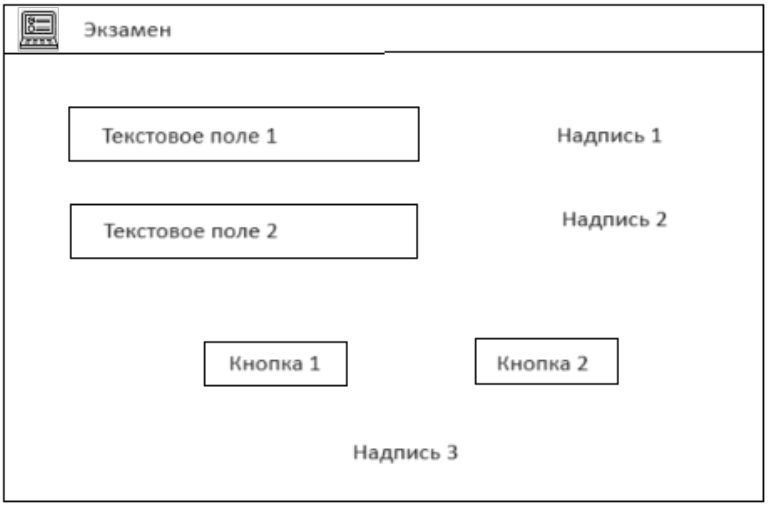
– **С отрицательными числами ничего не выйдет!**;
– **Нужно вводить только числа!**;
– **Жаль, но из этого треугольник не сделать**.

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

Задание №9
Написать программу в среде разработки VisualStudio, для следующей задачи:
Напишите программу, выводящая фамилию и инициалы с введенных в одну строку фамилии, имени и отчество, разделенные пробелом.
Пример:
Введите фамилию, имя и отчество:
Васильев Сергей Павлович
С.П. Васильев

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

Задание №10
Создайте проект и разместите в форме с заголовком "СКОРОСТЬ" следующие компоненты, как показано на рисунке:



Тексты надписей: для Надписи1 - "РАССТОЯНИЕ", для Надписи2 - "ВРЕМЯ", для Надписи3 - "СКОРОСТЬ: " должны быть крупными и хорошо видимыми в форме.
На Кнопке1 должен быть текст "ВЫЧИСЛИТЬ", на Кнопке2 – "ВЫХОД".
Текстовые поля первоначально должны быть пустыми
Для кнопки "ВЫЧИСЛИТЬ" создайте выполняющую следующие действия процедуру обработки нажатия на эту кнопку:
а) преобразование текстовых (строковых) значений свойства **Text** Текстового поля1 (для расстояния) и Текстового поля 2 (для времени) в соответствующие вещественные значения.
б) вычисление вещественного значения скорости по вещественным значениям расстояния и времени.
в) преобразование вещественного значения скорости в текстовое (строковое) значение и добавление этого значения к первоначальному тексту Надписи3 в виде текста справа.
Для кнопки "ВЫХОД" создайте процедуру обработки нажатия на эту кнопку для выхода из программы или, иначе говоря, для закрытия формы.
Отредактируйте созданную программу таким образом, чтобы она реагировала на ввод нулевого значения в Текстовое поле2 после нажатия кнопки ВЫЧИСЛИТЬ добавлением к первоначальному тексту Надписи3 текста 'ВРЕМЯ НЕ ДОЛЖНО БЫТЬ РАВНО 0!'

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

Задание №11
Написать программу в среде разработки VisualStudio, для следующей задачи:
Напишите рекурсивную функцию summation(n), которая будет принимать положительное целое число n и возвращать сумму чисел от 1 до n.

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

Задание №12
Написать программу в среде разработки VisualStudio, для следующей задачи:
Напишите программу, находящая среднее арифметическое всех чисел, которые

записаны в файле в столбик, и выводит результат в файл.

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

Задание №13
Написать программу в среде разработки VisualStudio, для следующей задачи: Имеется текстовый файл, в котором записана одна строка *Здравствуйте, дорогие ребята*. Приписать в конце строки восклицательный знак.

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

Задание №14
Написать программу в среде разработки VisualStudio, для следующей задачи: Определить комбинированный (структурный) тип для представления анкеты ребенка, состоящей из его имени, пола и роста. Ввести информацию по 5 детям. Вывести средний рост всех детей.

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

Задание №15
Написать программу в среде разработки VisualStudio, для следующей задачи: Создайте класс **Soda** (для определения типа газированной воды), принимающий 1 аргумент при инициализации (отвечающий за добавку к выбираемому лимонаду). В этом классе реализуйте метод **show_my_drink()**, выводящий на печать **Газировка и {ДОБАВКА}** в случае наличия добавки, а иначе отобразится следующая фраза: **Обычная газировка**.

Оценка	***Показатели оценки***

5	Программа работает верно, без ошибок.
4	Программа написана, работает с незначительными ошибками.
3	Программа написана, но не работает из-за ошибок.

Задание №16
Написать программу в среде разработки VisualStudio, для построенного алгоритма.

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

Задание №17
Написать программу в среде разработки VisualStudio, для построенного алгоритма, с использованием цикла While.

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

Дидактическая единица для контроля:
2.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования
Задание №1
Написать программу в среде разработки VisualStudio, для построенного алгоритма.

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

Задание №2
Написать программу в среде разработки VisualStudio, для построенного алгоритма, с использованием оператора выбора Switch.

Оценка	***Показатели оценки***

5	Программа работает верно, без ошибок.
4	Программа написана, работает с незначительными ошибками.
3	Программа написана, но не работает из-за ошибок.

Задание №3

Написать программу в среде разработки VisualStudio, для построенного алгоритма, с использованием цикла While.

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

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

Задание №2

Оформить код написанной программ в соответствии со стандартом кодирования.

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

Дидактическая единица для контроля:

2.7 Выполнять проверку, отладку кода программы

Задание №1

Выполнить проверку и отладку кода программы.

Оценка	***Показатели оценки***
5	Логика программы верна, работает корректно.
4	Логика программы верна, присутствуют незначительные ошибки.
3	Программа работает не корректно.

Задание №2
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