

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ АСТРАХАНСКОЙ ОБЛАСТИ  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
АСТРАХАНСКОЙ ОБЛАСТИ ВЫСШЕГО ОБРАЗОВАНИЯ  
«АСТРАХАНСКИЙ ГОСУДАРСТВЕННЫЙ  
АРХИТЕКТУРНО-СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ»  
(ГБОУ АО ВО «АГАСУ»)**

**Программа вступительного испытания  
«Основы программирования»**

**для поступающих в ГБОУ АО ВО «АГАСУ» по образовательным программам  
высшего образования – программам бакалавриата в 2024 году**

**АСТРАХАНЬ**

Программа вступительного испытания по дисциплине «Основы программирования» содержит задания по разделам: «Основные конструкции программирования», «Синтаксис конструкций программирования», «Рекурсия», «Структуры данных», «Концепция объектно-ориентированного программирования».

## **1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРОВЕДЕНИЮ ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ**

Вступительные испытания по дисциплине «Основы программирования» проводятся в письменной форме.

Абитуриент получает экзаменационный билет, содержащий два теоретических вопроса и одну задачу по программированию. Абитуриент излагает содержание вопросов и решение задачи письменно.

Длительность экзамена - 2 часа (120 мин). Систем оценивания – сто балльная. Пользоваться справочными материалами любого рода во время подготовки запрещается.

## **2. КРИТЕРИИ ОЦЕНИВАНИЯ ЗНАНИЙ**

Работа состоит из двух частей. Часть 1 включает два теоретических вопроса. Каждый правильный ответ оценивается в 25 баллов, таким образом, максимальное количество баллов за этот вид работы - 50. Часть 2 состоит из одной задачи. Правильный ответ задачи оценивается в 50 баллов.

Баллы, полученные за все выполненные задания, суммируются. Максимальное количество баллов - 100.

## **3. СОДЕРЖАНИЕ ПРОГРАММЫ**

### **3.1. Основные конструкции программирования**

Понятие среды (системы) программирования. Основные компоненты среды программирования, их назначение. Понятие цикла разработки программы на некотором ЯП: назначение этапов цикла и действия разработчика. Классификация ошибок в программе: синтаксические, семантические ошибки, ошибки времени выполнения. Понятие отладчика: назначение и основные функции. Понятие директивы компилятора.

### **3.2. Синтаксис конструкций программирования**

Понятие константы в ЯП. Объявление и использование констант (включая также типизированные константы). Понятие выражения в ЯП. Классификация и приоритет операций. Построение и вычисление выражений. Понятие типа данных в ЯП, классификация, примеры. Понятие эквивалентности типов данных, структурная и именная эквивалентность типов. Понятие совместимости типов, совместимость по присваиванию. Преобразование типов. Понятие оператора в ЯП. Классификация операторов. Примеры операторов. Понятие структурного программирования. Теорема о

структурном программировании. Понятие подпрограммы в ЯП. Виды подпрограмм. Спецификация подпрограммы. Формальные и фактические параметры подпрограммы: определение, назначение, установление взаимного соответствия. Взаимно рекурсивные подпрограммы. Способы передачи фактических параметров в подпрограмму. Виды формальных параметров. Разработка подпрограмм: выбор вида формальных параметров. Понятие модульного программирования, модульная структура программы. Структура программы на ЯП высокого уровня. Понятия определяющего и использующего вхождения идентификатора, области действия и видимости декларации. Локальные и глобальные переменные. Понятие побочного эффекта подпрограммы. Распределение памяти программы: сегмент данных, сегмент стека, куча. Файловые типы данных. Классификация файловых типов. Стандартные подпрограммы работы с файлами. Ссылочные типы данных и указатели. Статические и динамические переменные программы. Стандартные подпрограммы работы с указателями. Библиотеки в ЯП. Структура и назначение компонент библиотеки

### **3.3. Рекурсия**

Понятие рекурсии и итерации. Структура рекурсии. Реализация механизма рекурсии. Прямой и обратный ход рекурсии. Рекурсивные алгоритмы с заглядыванием вперед и с возвратом.

### **3.4. Структуры данных**

Понятие структуры данных. Логическая и физическая структура данных. Классификация структур данных. Последовательные списки: стек, очередь, дек. Связные списки: однонаправленный список, двунаправленный список, циклический список. Понятие графа. Дерево. Бинарное дерево поиска. Прямой, обратный, концевой обход дерева. Алгоритмы добавления и удаления вершины из бинарного дерева поиска

### **3.5. Концепция объектно-ориентированного программирования**

Принципы ООП. Понятие объекта, характеристика объекта, видимость атрибутов и методов, синтаксис. Раннее и позднее связывание

## **4. ЛИТЕРАТУРА**

1. Златопольский, Д. М. Программирование: типовые задачи, алгоритмы, методы : учебное пособие : [12+] / Д. М. Златопольский. – 4-е изд. (эл.). – Москва : Лаборатория знаний, 2020. – 226 с. : ил. – Режим доступа: по подписке. URL: <https://biblioschool.ru/index.php?page=book&id=222873> (дата обращения: 26.10.2021). – Библиогр. в кн. – ISBN 978-5-00101-789-9. – Текст : электронный.

2. Нагаева, И. А. Алгоритмизация и программирование: практикум : учебное пособие : [12+] / И. А. Нагаева, И. А. Кузнецов. – Москва ; Берлин :

Директ-Медиа, 2019. – 168 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioschool.ru/index.php?page=book&id=570287> (дата обращения: 26.10.2021). – Библиогр.: с. 163-164. – ISBN 978-5-4499-0314-3. – DOI 10.23681/570287. – Текст : электронный.

3. Основы программирования (язык C) [Электронный ресурс]: Учеб. пособие / Двойнишников С.В. – Новосибирск : РИЦ НГУ, 2018. - <http://www.studentlibrary.ru/book/ngu006.html> (ЭБС «Консультант студента»).

4. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для СПО / В. В. Трофимов, Т. А. Павловская ; под ред. В. В. Трофимова. — М. : Издательство Юрайт, 2019. — 137 с.

5. Гниденко, И. Г. Технология разработки программного обеспечения : учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М. : Издательство Юрайт, 2017. — 235 с.

6. Кудрина, Е. В. Основы алгоритмизации и программирования на языке `c#` : учеб. пособие для СПО / Е. В. Кудрина, М. В. Огнева. — М. : Издательство Юрайт, 2019. — 322 с.

7. Федоров, Д. Ю. Программирование на языке высокого уровня `python` : учеб. пособие для СПО / Д. Ю. Федоров. — М. : Издательство Юрайт, 2019. — 126 с.

8. Мамонова, Т. Е. Информационные технологии. Лабораторный практикум : учеб. пособие для СПО / Т. Е. Мамонова. — М. : Издательство Юрайт, 2019. — 178 с.

*Теоретические вопросы*

1. Алгоритмы и их свойства. Способы задания и описания алгоритмов. Блок-схема алгоритма. Основные алгоритмические структуры: линейная, ветвления, циклы.
2. Программирование. Основные положения.
3. Величины. Постоянные и переменные величины. Аргументы и результаты. Типы величин.
4. Языки программирования. Алфавит, синтаксис, семантика.
5. Стандартные типы данных. Константы и переменные.
6. Целые типы данных и операции над ними.
7. Вещественные типы данных и операции над ними. Формы записи вещественных констант с фиксированной и плавающей точкой.
8. Стандартные функции. Арифметические выражения.
9. Логический тип данных. Описание и операции над данными логического типа. Отношения. Упорядоченные типы данных. Логические выражения.
10. Символьный тип данных. Стандартные функции с символьными значениями и аргументами.
11. Строковый тип данных. Описание и операции над данными этого типа. Особенности применения.
12. Перечислимые типы данных. Описание и особенности применения.
13. Оператор присваивания. Согласование типов данных.
14. Операторы ввода и вывода. Форматный вывод.
15. Оператор бинарного ветвления (условный) полный и неполный.
16. Оператор множественного ветвления (выбора).
17. Оператор цикла с предусловием. Применение при реализации итерационных методов.
18. Оператор цикла с постусловием. Применение при реализации итерационных методов.
19. Циклы с конечным числом повторений. Вложенные циклы.
20. Организация циклических процессов. Примеры рекуррентных алгоритмов.
21. Одномерные массивы. Их описание. Ввод, вывод. Примеры задач.
22. Двумерные массивы. Их описание. Ввод, вывод. Примеры задач.
23. Процедуры и функции. Их описание и применение.
24. Процедуры с параметрами и без параметров. Их описание и применение.
25. Алгоритмы внутренней сортировки. Сортировка выбором.
26. Алгоритмы внутренней сортировки. Сортировка вставкой.
27. Алгоритмы внутренней сортировки. Пузырьковая сортировка.
28. Этапы подготовки и решения задач на ЭВМ
29. Стили программирования. Процедурное программирование.
30. Стили программирования. Функциональное и логическое программирование.
31. Стили программирования. Объектно-ориентированное программирование.
32. Жизненный цикл программного средства.
33. Понятие качества программного средства.
34. Модульное программирование. Основные характеристики программного модуля.
35. Метод восходящей разработки структуры программы.
36. Метод нисходящей разработки структуры программы.
37. Отладка программного средства. Принципы и виды отладки.
38. Сущность объектного подхода к разработке программных средств. Его отличие от функционального подхода.
39. Объектный подход к разработке программных средств. Объектная модель
40. Принципы объектно-ориентированного программирования. Инкапсуляция.
41. Принципы объектно-ориентированного программирования. Наследование

## 42. Принципы объектно-ориентированного программирования. Полиморфизм

### *Примерный текст задачи*

1. В матрице целых чисел найти наибольший (максимальный) по модулю элемент. Вывести на экран его индексы и значение.
2. Написать функцию, которая возвращает среднее арифметическое двух переданных ей аргументов (параметров).
3. Все элементы массива поделить на значение наибольшего элемента этого массива.
4. Вводится натуральное число. Найти сумму четных цифр, входящих в его состав.
5. Дан массив, содержащий положительные и отрицательные числа. Заменить все элементы массива на противоположные по знаку.
6. В массиве найти минимальный и максимальный элементы, поменять их местами.
7. Найти сумму тех элементов массива, которые одновременно имеют четные и отрицательные значения.
8. Найти (вычислить) сумму и произведение элементов матрицы (двумерного массива).
9. Найти количество положительных элементов, лежащих на главной диагонали квадратной матрицы
10. Из двух чисел с разной четностью вывести на экран нечетное число.
11. Пользователь вводит целое число. Надо вывести на экран сколько в этом числе цифр и положительное оно или отрицательное. Например, "это однозначное положительное число".
12. Вывести элементы числового массива, которые больше, чем элементы, стоящие перед ними.
13. Все элементы массива поделить на значение наименьшего элемента этого массива.
14. Найти номер и значение первого положительного элемента массива.
15. Найти максимальный элемент для каждого столбца матриц размерностью  $N \times N$ .
16. Создайте из четных символов строки *str1* строку *str2* и из нечетных символов - строку *str3*.
17. Дана матрица (двумерный массив). Поменять местами две любые ее строки.
18. В одномерном массиве найти минимальный и максимальный элементы. Вычислить их разность.
19. Написать функцию, которая возвращает среднее арифметическое двух переданных ей аргументов (параметров).
20. Найти количество отрицательных элементов, лежащих на главной диагонали квадратной матрицы
21. Написать программу, подсчитывающую количество четных и нечетных цифр в числе.
22. В двумерном массиве целых чисел определить, сколько раз в нем встречается элемент со значением  $X$ .
23. Определить какое из трех, введенных пользователем, чисел максимальное и вывести его на экран.
24. Требуется написать программу, вычисляющую значение какой-либо функции  $y = f(x)$ . Допустим, такой:  
$$y = x - 2, \text{ если } x > 0,$$
$$y = 0, \text{ если } x = 0,$$
$$y = |x|, \text{ если } x < 0.$$
25. Всем известна прямоугольная (декартова) система координат, в которой две перпендикулярные оси делят плоскость на четверти. В первую четверть попадают точки, у которых обе координаты ( $x$  и  $y$ ) больше нуля. Во вторую:  $x < 0, y > 0$ ; третью:  $x < 0, y < 0$ ;

четвертую:  $x > 0, y < 0$ . Требуется написать программу, определяющую по координатам точки, в какой четверти она находится.