

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)



УТВЕРЖДАЮ
И.о. ректора
Ю.Н. Старилев
16.01.2026 г.

ПРОГРАММА ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ ПРИ ПРИЕМЕ
НА ОБУЧЕНИЕ ПО ПРОГРАММАМ БАКАЛАВРИАТА И СПЕЦИАЛИТЕТА
ОСНОВЫ ИНФОРМАТИКИ, АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

**(для поступающих на базе профильного
среднего профессионального образования)**

Программа разработана на основе федеральных государственных образовательных стандартов среднего профессионального образования по специальностям

- 09.02.01 Компьютерные системы и комплексы
- 09.02.06 Сетевое и системное администрирование
- 09.02.07 Информационные системы и программирование
- 09.02.08 Интеллектуальные интегрированные системы
- 09.02.09 Веб-разработка
- 09.02.10 Разработка компьютерных игр, дополненной и виртуальной реальности
- 09.02.11 Разработка и управление программным обеспечением
- 09.02.12 Техническая эксплуатация и сопровождение информационных систем
- 09.02.13 Интеграция решений с применением технологий искусственного интеллекта
- 10.02.04 Обеспечение информационной безопасности телекоммуникационных систем
- 10.02.05 Обеспечение информационной безопасности автоматизированных систем

ТРЕБОВАНИЯ К УРОВНЮ ПОДГОТОВКИ АБИТУРИЕНТОВ

В ходе вступительного испытания абитуриент должен продемонстрировать следующие знания и умения.

Знания:

- основные принципы математической логики, теории множеств и теории алгоритмов;
- формулы алгебры высказываний;
- методы минимизации алгебраических преобразований;
- основы языка и алгебры предикатов;
- основные принципы теории множеств;
- принципы построения позиционных систем счисления; развёрнутую форму записи чисел; алгоритмы перевода чисел между системами счисления; правила выполнения арифметических операций в различных системах счисления; связь между двоичной, восьмеричной и шестнадцатеричной системами;
- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;

- 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.

Умения:

- применять логические операции, формулы логики, законы алгебры логики;
- формулировать задачи логического характера и применять средства математической логики для их решения;
- переводить целые и дробные числа между различными системами счисления; выполнять арифметические операции в двоичной, восьмеричной и шестнадцатеричной системах; применять методы быстрого перевода между системами с кратными основаниями;
- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

ФОРМА ПРОВЕДЕНИЯ И СТРУКТУРА ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ

Вступительное испытание по дисциплине «Основы информатики, алгоритмизации и программирования» проводится в письменной форме. На выполнение экзаменационной работы отводится 3 часа (180 минут).

Экзаменационная работа содержит 10 заданий. Решение каждого задания оценивается от 0 до 10 баллов с шагом в 1 балл на основе разработанной системы критериев оценивания. Баллы, полученные за выполненные задания, суммируются. Максимальный балл за экзаменационную работу равен 100.

СОДЕРЖАНИЕ ПРОГРАММЫ ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ

Тема 1. Основы математической логики.

- Понятие высказывания. Основные логические операции.
- Формулы логики. Таблица истинности и методика её построения.
- Законы логики. Равносильные преобразования.
- Понятие булевой функции. Способы задания ДНФ, КНФ.

- Операция двоичного сложения и её свойства. Многочлен Жегалкина.
- Основные классы функций. Полнота множества. Теорема Поста.

Тема 2. Элементы теории множеств.

- Общие понятия теории множеств. Способы задания. Основные операции над множествами и их свойства.
- Мощность множеств. Графическое изображение множеств на диаграммах Эйлера-Венна. Декартово произведение множеств.

Тема 3. Системы счисления.

- Системы счисления. Развёрнутая запись целых и дробных чисел в позиционных системах счисления.
- Алгоритм перевода целого числа из P -ичной системы счисления в десятичную. Алгоритм перевода целого числа из десятичной системы счисления в P -ичную.
- Двоичная, восьмеричная и шестнадцатеричная системы счисления, перевод чисел между этими системами.
- Арифметические операции в позиционных системах счисления.

Тема 4. Типы данных.

- Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.

Тема 5. Операторы языка программирования.

- Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.
- Условный оператор. Оператор выбора.
- Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.
- Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.
- Структурированный тип данных – множество. Операции над множествами.
- Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа.

Тема 6. Процедуры и функции.

- Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.
- Рекурсия. Программирование рекурсивных алгоритмов.

Тема 7. Основные принципы объектно-ориентированного программирования (ООП).

- Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.
- Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
- Классы объектов. Компоненты и их свойства.

ОБРАЗЕЦ КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНОГО МАТЕРИАЛА

Задание 1. Переведите следующие числа:

- а) 10110_2 в десятичную систему счисления;
- б) 157_{10} в двоичную систему счисления;
- в) 64_{10} в восьмеричную систему счисления.

Запишите подробное решение.

Задание 2. Выполните сложение в двоичной системе счисления:

$$1110_2 + 1101_2$$

Запишите подробное решение. Ответ дайте в двоичной системе счисления.

Задание 3. Постройте таблицу истинности для логического выражения:

$$F = (A \vee B) \rightarrow (\neg A \wedge C)$$

Определите, при каких значениях переменных А, В, С формула принимает значение истина.

Задание 4. Определите значение переменной x после выполнения программы, представленной ниже (программа записана на четырёх языках программирования). Подробно опишите ход работы программы.

Python	Паскаль
<pre>a = 5 b = 12 c = 8 if a < b: if b > c: x = a + c else:</pre>	<pre>var a, b, c, x: integer; begin a := 5; b := 12; c := 8; if a < b then if b > c then x := a + c else</pre>

<pre> x = b - a else: x = c * 2 print(x) </pre>	<pre> x := b - a else x := c * 2; write(x); end. </pre>
C++	Java
<pre> #include <iostream> using namespace std; int main() { int a, b, c, x; a = 5; b = 12; c = 8; if (a < b) { if (b > c) { x = a + c; } else { x = b - a; } } else { x = c * 2; } cout << x << endl; return 0; } </pre>	<pre> public class Task { public static void main(String[] args) { int a, b, c, x; a = 5; b = 12; c = 8; if (a < b) { if (b > c) { x = a + c; } else { x = b - a; } } else { x = c * 2; } System.out.println(x); } } </pre>

Задание 5. Даны три множества:

$$A = \{1, 2, 3, 4, 5\},$$

$$B = \{3, 4, 5, 6, 7\},$$

$$C = \{5, 6, 7, 8, 9\}.$$

Найдите:

1) $A \cap B$ (пересечение множеств A и B)

2) $A \cup C$ (объединение множеств A и C)

3) $B \setminus A$ (разность множеств B и A)

4) $(A \cup B) \cap C$

Изобразите множества A , B , C на диаграмме Эйлера-Венна и покажите на ней результат операции $(A \cup B) \cap C$.

Задание 6. Автомат получает на вход трёхзначное число. По этому числу строится новое число по следующим правилам:

- 1) Перемножаются первая и вторая цифры, а также вторая и третья цифры исходного числа.
- 2) Полученные два числа записываются друг за другом в порядке неубывания (без разделителей).

Пример: Исходное число: 234. Произведения: $2 \cdot 3 = 6$; $3 \cdot 4 = 12$. Результат: 612.

Найдите наименьшее трёхзначное число, при обработке которого автомат выдаст число 1215. Обоснуйте ответ, продемонстрировав всю цепочку рассуждений.

Задание 7. Упростите логическое выражение, используя законы алгебры логики:

$$F = (A \wedge B) \vee (A \wedge \neg B) \vee (\neg A \wedge B)$$

Покажите все промежуточные шаги упрощения и укажите, какие законы Вы применяли.

Задание 8. Определите, сколько раз выполнится тело цикла и какое значение будет иметь переменная k после выполнения следующего фрагмента программы:

Python	Паскаль
<pre>k = 0 i = 10 while i > 0: k = k + 1 i = i - 2</pre>	<pre>k := 0; i := 10; while i > 0 do begin k := k + 1; i := i - 2; end;</pre>
C++	Java
<pre>int k = 0; int i = 10; while (i > 0) { k = k + 1; i = i - 2; }</pre>	<pre>int k = 0; int i = 10; while (i > 0) { k = k + 1; i = i - 2; }</pre>

Требуется:

- 1) Определить количество итераций цикла.
- 2) Определить итоговое значение переменной k.
- 3) Представить процесс выполнения в виде таблицы со значениями i и k на каждой итерации.

Задание 9. Напишите программу на любом широко известном языке программирования (Pascal, Python, Java, C/C++, C#), которая:

- 1) Запрашивает у пользователя ввод целого числа N.
- 2) Проверяет, что N находится в диапазоне $1 \leq N \leq 100$.
- 3) Если N не удовлетворяет условию, выводит сообщение об ошибке и завершает работу.
- 4) Вычисляет сумму всех чётных чисел от 1 до N.
- 5) Выводит результат на экран.

Пример работы программы:

```
Введите N (1-100): 10
Сумма чётных чисел от 1 до 10: 30
```

Задание 10. Дан массив из 10 элементов:

[12, -5, 8, -3, 15, 0, -7, 21, -9, 4]

Напишите программу на любом широко известном языке программирования (Pascal, Python, Java, C/C++, C#), которая:

- 1) Находит максимальный и минимальный элементы массива.
- 2) Подсчитывает количество положительных, отрицательных и нулевых элементов.
- 3) Вычисляет среднее арифметическое всех положительных элементов.
- 4) Создаёт новый массив, в котором все отрицательные числа заменены на 0.
- 5) Выводит исходный и новый массивы.

Приведите полный текст программы с комментариями.

Критерии оценивания выполнения заданий

Балл	Критерии
10	<p>Полное правильное решение</p> <ul style="list-style-type: none"> – Задание решено верно и полностью – Для вычислительных заданий: приведены все необходимые этапы решения, промежуточные вычисления, обоснования – Для программных заданий: программа написана корректно, содержит необходимые комментарии, обрабатывает все требуемые случаи – Получен правильный ответ – Решение оформлено логично и читаемо – Отсутствуют недочёты
9	<p>Решение с незначительным недочётом</p> <ul style="list-style-type: none"> – Задание решено верно – Допущена одна незначительная описка (опечатка в переменной, арифметическая ошибка в промежуточных вычислениях) – Метод решения/алгоритм выбран правильно – Для программ: логика корректна, но есть минорная синтаксическая неточность, не влияющая на работоспособность – Или: недочёт в оформлении (отсутствие одного комментария, незначительное нарушение стиля кода)
8	<p>Решение с единичной ошибкой</p> <ul style="list-style-type: none"> – Метод решения/алгоритм выбран верно – Допущена одна ошибка в вычислениях или логике, приводящая к неверному ответу – Дальнейший ход решения логичен и последователен – Для программ: алгоритм верен, но допущена ошибка в реализации одного фрагмента (неверное условие, ошибка в формуле) – Или: пропущен один из этапов решения/требований задачи, но основная часть выполнена корректно
7	<p>Решение с несколькими ошибками</p> <ul style="list-style-type: none"> – Метод решения/алгоритм выбран правильно – Допущены 2-3 вычислительные или логические ошибки – Общая логика решения прослеживается и верна

	<ul style="list-style-type: none"> – Для программ: основной алгоритм понятен и в целом корректен, но есть 2-3 ошибки в деталях реализации – Или: решение почти завершено, но финальный этап не выполнен или выполнен с ошибкой
6	<p>Верный подход с существенной ошибкой</p> <ul style="list-style-type: none"> – Выбран правильный метод/алгоритм – Начальные этапы решения выполнены верно (50-70% задания) – Допущена существенная ошибка в реализации, искажающая результат – Для программ: структура программы правильная, основные блоки присутствуют, но один из ключевых фрагментов реализован неверно – Или: не учтён важный случай/требование (например, проверка диапазона, обработка граничных значений)
5	<p>Половина решения выполнена верно</p> <ul style="list-style-type: none"> – Правильно выбран метод/подход – Верно выполнено около 50% решения – Дальнейшее решение содержит принципиальные ошибки или отсутствует – Для программ: реализована примерно половина требуемого функционала, остальное отсутствует или содержит грубые ошибки – Или: задача решена частично (например, выполнено 2 из 4 пунктов)
4	<p>Решение начато верно</p> <ul style="list-style-type: none"> – Правильно определён метод решения/выбран алгоритм – Выполнены начальные шаги (30-50% решения) – Продемонстрировано понимание подхода к задаче – Для программ: определена структура программы, объявлены переменные, начата реализация основной логики – Дальнейшее решение содержит ошибки или не завершено
3	<p>Частичное понимание задачи</p> <ul style="list-style-type: none"> – Сделана попытка решения с использованием релевантного подхода – Выполнено менее 30% решения – Присутствуют верные элементы (формулы, таблицы истинности, объявления переменных, фрагменты кода) – Для программ: записан каркас программы или отдельные корректные строки кода
2	<p>Минимальное продвижение</p>

	<ul style="list-style-type: none"> – Записаны релевантные формулы, определения или синтаксические конструкции – Сделаны начальные действия без существенного продвижения – Продемонстрировано понимание того, что требуется – Для программ: записаны заголовок программы, объявления или отдельные операторы без связной логики – Или: задача переформулирована в виде алгоритма на естественном языке
1	<p>Попытка решения</p> <ul style="list-style-type: none"> – Попытка решения – Сделана попытка приступить к решению – Записаны какие-либо формулы, операторы, относящиеся к теме – Понимание метода решения не продемонстрировано – Для программ: записаны разрозненные строки кода без понимания общей структуры
0	<p>Отсутствие решения</p> <ul style="list-style-type: none"> – Отсутствие решения – Задание не выполнялось – Записано содержимое, не относящееся к заданию – Решение полностью неверно и не содержит элементов правильного подхода

Примечания:

- Синтаксические ошибки, не влияющие на понимание алгоритма (пропущенная точка с запятой, неправильное написание имени функции вывода), снижают оценку на 1 балл (при повторении однотипных ошибок далее балл не снижается)
- Работоспособность кода не может быть проверена в письменной форме, поэтому оценивается читаемость алгоритма и корректность логики
- Псевдокод вместо программы на конкретном языке допустим и оценивается, но максимальный балл в этом случае равен 7 (так как не выполнено требование написать программу на языке программирования).

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Семакин, И. Г. Основы алгоритмизации и программирования: учебное издание / Семакин И.Г., Шестаков А. П. — Москва : Академия, 2024. — 304 с.
2. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская. — 4-е изд. — Москва : Издательство Юрайт, 2025. — 108 с. — (Профессиональное образование). — ISBN 978-5-534-20429-2.
3. Якимов, С. П. Алгоритмизация и программирование : учебник для среднего профессионального образования / С. П. Якимов. — Москва : Издательство Юрайт, 2025. — 342 с. — (Профессиональное образование). — ISBN 978-5-534-19661-0.
4. Кудрявцева, И. А. Программирование: комбинаторная логика : учебник для среднего профессионального образования / И. А. Кудрявцева, М. В. Швецкий. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 524 с.
5. Гниденко, И. Г. Технология разработки программного обеспечения : учебник для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 248 с. — (Профессиональное образование). — ISBN 978-5-534-18131-9.
6. Скорубский, В. И. Математическая логика : учебник и практикум для среднего профессионального образования / В. И. Скорубский, В. И. Поляков, А. Г. Зыков. — Москва : Издательство Юрайт, 2025. — 211 с. — (Профессиональное образование). — ISBN 978-5-534-11631-1.
7. Гашков, С. Б. Дискретная математика : учебник и практикум для среднего профессионального образования / С. Б. Гашков, А. Б. Фролов. — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 530 с. — (Профессиональное образование). — ISBN 978-5-534-17715-2.
8. Гисин, В. Б. Дискретная математика : учебник и практикум для среднего профессионального образования / В. Б. Гисин. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 468 с. — (Профессиональное образование). — ISBN 978-5-534-16754-2.
9. Судоплатов, С. В. Дискретная математика : учебник и практикум для среднего профессионального образования / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 279 с. — (Профессиональное образование). — ISBN 978-5-534-11632-8.