

ПРОГРАММА
профильного вступительного испытания по дисциплине
«Основы информационных технологий»

(для выпускников учреждений среднего специального образования по специальностям «Программируемые мобильные системы», «Программное обеспечение информационных технологий», «Тестирование программного обеспечения», «Вычислительные машины, системы и сети», «Электронные вычислительные средства»)

I. Общие указания

Программа профильного вступительного испытания по дисциплине «Основы информационных технологий» рассчитана на поступающих в 2017 году на механико-математический факультет БГУ для получения высшего образования I ступени по специальности 1-31 03 08 «Математика и информационные технологии» (сокращенная форма).

Программа разработана на основе типовых учебных программ по дисциплинам «Основы алгоритмизации и программирования», «Структуры и алгоритмы обработки данных», «Системное программное обеспечение», «Компьютерные сети», «Технология разработки программного обеспечения» и «Программные средства создания интернет-приложений» для учреждений, реализующих образовательную программу среднего специального образования.

Цель профильного вступительного испытания:

определить уровень знаний абитуриентов по наиболее важным разделам информатики как основы подготовки специалистов с высшим образованием в сфере современных информационных технологий.

Задачи профильного вступительного испытания:

- выяснить объем знаний, умений и навыков в соответствии с содержанием программы вступительного испытания;
- оценить знания абитуриентов, используя критерии оценки уровня подготовки абитуриентов;
- осуществить качественный отбор абитуриентов.

Абитуриент должен знать на уровне представления:

- значение информационных технологий в современном мире;
- эволюцию языков программирования, их классификацию;
- структуру программного обеспечения и систем программирования;
- современные технологии разработки программного обеспечения и перспективы их развития;
- методы и средства программирования с использованием существующих технологий;
- статические и динамические структуры данных;
- зависимость эффективности алгоритмов от способов представления данных;
- способы обмена данными в информационных системах;
- современную методологию и средства математического моделирования.

Абитуриент должен знать на уровне понимания:

- принципы построения эффективных алгоритмов;
- основы структурного проектирования программ;
- общие принципы построения алгоритмов, способы описания алгоритмов;
- методы разработки программ, основные элементы языка программирования;
- принципы программирования на процедурно-ориентированном языке;
- виды подпрограмм, методику разработки библиотек подпрограмм;
- понятие информации и информационных технологий;
- суть и основные этапы технологического процесса обработки информации;
- технологию сбора, хранения, передачи, обработки и представления информации;
- способы обработки текстовой и числовой информации;
- гипертекстовые способы хранения и представления информации;
- мультимедийные технологии обработки и представления информации;
- средства и методы визуального программирования;
- технологии создания интернет-приложений.

Абитуриент должен уметь:

- разрабатывать алгоритм решения задачи;
- разрабатывать программы, проводить их отладку, тестирование и верификацию;
- использовать стандартное математическое обеспечение;
- использовать разработанные библиотеки подпрограмм;
- осуществлять выбор необходимых аппаратных, информационных и программных средств для обеспечения процесса обработки информации;
- производить установку, адаптацию и эксплуатацию программного обеспечения;
- использовать возможности компьютерных сетей;
- проводить исследование предметной области решаемой задачи и анализировать полученный результат;
- обосновывать выбор языка программирования и среды разработки программы;
- осуществлять разработку приложений, их отладку, тестирование, верификацию и документирование с использованием возможностей выбранных средств;
- применять средства автоматизации программирования, реализовывать возможности визуального и объектно-ориентированного программирования.

Программа вступительного испытания включает темы, отражающие сведения об устройстве компьютера, операционных системах, видах алгоритмов, прикладном программном обеспечении, технологиях программирования, основах веб-технологий, без чего невозможна подготовка квалифицированных специалистов в области современных информационных технологий.

II. Требования к профильному вступительному испытанию

Содержание программы вступительных испытаний

Устройство компьютера. Компьютерные коммуникации и сети

Типы компьютеров.

Основные устройства компьютера.

Архитектура компьютера. Принципы работы компьютера.

Представление информации в компьютере.

Компьютерные сети.

Адресация в IP-сетях. Система доменных имен DNS.

Протоколы TCP/IP.

Администрирование пользователей и групп в сети.

Операционные системы. Системное программное обеспечение

Архитектура вычислительных систем. Принципы функционирования вычислительных систем.

Операционные системы (ОС), функции и режимы их работы.

Обработка прерываний.

Структура ОС Windows.

Статическая и динамическая компоновка. DLL-библиотеки и их структура. Динамическое связывание в процессе загрузки и во время выполнения программ.

Организация управления данными. Файловые системы.

Основы архитектуры ОС Unix, Linux. Особенности и возможности ОС Unix, Linux.

Алгоритмы

Алгоритмы, виды алгоритмов.

Операции, элементарные операции.

Основные структуры данных.

Трудоёмкость алгоритмов. Принципы оценки трудоёмкости комбинаторных алгоритмов.

Пакеты. Прикладное программное обеспечение

Информация. Способы представления информации.

Обработка текстовой информации. Текстовый процессор Word.

Обработка числовой и математической информации. Пакеты Mathcad, Mathematica или Maple.

Обработка графической информации. Пакеты Adobe PhotoShop, CorelDraw.

Обработка мультимедийной и аудиовизуальной информации.

Программное обеспечение обработки экономической и деловой информации. Электронные таблицы Excel.

Система презентаций PowerPoint.

СУБД Access.

Технологии программирования

Системы программирования. Интегрированные средства разработки.

Структура современной системы программирования.

Жизненный цикл разработки программного обеспечения (ПО).

Понятие мобильности и переносимости ПО. Структура переносимого ПО.

Стандарты переносимости.

Объектно-ориентированная технология разработки ПО.

Понятие о языке моделирования UML. Виды диаграмм.

Базовые принципы объектного программирования.

Принципы быстрой разработки программ.

Понятие об унифицированном процессе разработки (UP, RUP).

Виды отношений между классами и объектами.

Абстрактные классы и интерфейсы. Назначение интерфейсов.

Понятие о каскадном и итеративном процессах разработки программ.

Понятие о трехуровневых и многоуровневых приложениях.

Понятие о шаблонах (паттернах) проектирования.

Понятие о тестировании. Модульное тестирование.

Способы хранения, обработки и передачи информации. Базы данных

Файлы.

Потоки.

Электронные таблицы.

Базы данных. Виды баз данных.

Реляционные базы данных.

Структурированный язык запросов SQL.

Архивация данных.

Безопасность и защита данных.

Основы веб-технологий и интернет-приложений

Глобальная сеть Интернет. Сервисы.

Система адресации. DNS-серверы.

Межсетевой протокол IP.

Протокол TCP. Состояния TCP-сеанса.

Системы электронной почты.

Интернет и Веб. Веб-технологии. Ресурсы Веб.

HTTP-сообщения. Заголовки HTTP. Сессия HTTP.

Язык HTML. Тэги. Контейнеры.

Скрипты на стороне клиента. Встраивание скриптов в документ HTML.

Технология Dynamic HTML. События. Обработчики событий.

Разработка приложений в архитектуре клиент-сервер.

Формы в документе HTML. Основные действия пользовательского агента при отправке формы. Передача информации от сервера веб-приложению CGI.

Сценарии стороны сервера. Взаимодействие компонентов программного обеспечения при запуске сценариев стороны сервера.

Языки программирования на стороне сервера.

Задача публикации баз данных в Интернет. Способы публикации баз данных в Интернет.

Примеры экзаменационных заданий

Экзаменационное задание № 1

Задача 1. Что будет выведено в результате выполнения команды
`printf("%x dreams never come true.\n", 2989);`

Задача 2. Алгоритм вычисления значения функции $f(n)$, где n – натуральное число, при $n \geq 2$ задан соотношением $f(n) = f(n-1) + 3f(n-2)$. Известно также, что $f(1) = 0$ и $f(2) = 2$. Найдите $f(6)$.

Задача 3. Сколько оперативной памяти должен занимать тип `type`, чтобы в результате следующего фрагмента кода переменная `c` хранила `-32767`?

```
type a = 32767;
```

```
type b = 1;
```

```
type c = 1;
```

```
c = a + b;
```

Задача 4. Дайте определение языка UML. Классифицируйте диаграммы UML.

Задача 5. Что будет выведено в результате следующего фрагмента кода?

```
const int n = 7;
```

```
char array[n] = {'a', 'b', 'c', 'd', 'e', 'f', 'g'};
```

```
for(int i = 1; i < n; ++i)
```

```
    array[i] = array[n-i];
```

```
for(int i = 0; i < n; ++i)
```

```
    cout << array[i];
```

Задача 6. Опишите на одном из языков программирования алгоритм, для нахождения суммы всех трехзначных чисел, не делящихся на 7.

Задача 7. Дайте определение рефакторинга и укажите основные причины его использования.

Задача 8. Опишите основные признаки плохого кода (*code smells*), требующего оптимизации понимания работы программы.

Задача 9. Маской подсети называется 32-разрядное двоичное число, которое определяет, какая часть IP-адреса компьютера относится к адресу сети, а какая часть IP-адреса определяет адрес компьютера в подсети. В маске подсети старшие биты, отведенные в IP-адресе компьютера для адреса сети, имеют значение 1; младшие биты, отведенные в IP-адресе компьютера для адреса компьютера в подсети, имеют значение 0.

Если маска подсети — это 255.255.255.192 и IP-адрес компьютера в сети — 10.18.134.220, то чему равен номер компьютера в сети?

Задача 10. Дан следующий фрагмент кода:

```
int main()
```

```
{
```

```
int a = 1, b = 1;
```

```
int x, y;
```

```
scanf("%d%d", &x, &y); // пользователь вводит два числа
```

```
printf("%d %d", a, b);
```

```
for(int i = 3; i <= 10; ++i)
```

```

    {
    int c = x * a + y * b;
    printf("%d ", c);
    a = b;
    b = c;
    }
printf("\n");
return 0;
}

```

Каков будет результат, если пользователь введет числа 1 и 2?

Задача 11. В следующем коде измените ровно один знак так, чтобы напечаталось 20 звездочек:

```

int n = 20;
for (int i = 0; i < n; i--) printf( "*" );

```

Предложите три варианта ответов.

Задача 12. На вершине лестницы, состоящей из восьми ступенек, находится робот, который начинает спускаться к основанию лестницы. Робот может шагнуть на следующую ступеньку, на ступеньку через одну или через две. Определите число всевозможных «маршрутов» робота с вершины лестницы к основанию.

Задача 13. Дан следующий фрагмент кода:

```

#define n 8
int A[n];
int B[n];
void fun(int left, int right)
{
if(left < right)
{
int mid = (left + right) / 2;
fun(left, mid);
for(int i = left, j = mid + 1; i <= mid; ++i, ++j)
{
//printf("(%d, %d)", i, j);
B[i] = A[i] + A[j];
B[j] = A[i] - A[j];
}
for(int i = left; i <= right; ++i)
{
A[i] = 2 * B[i];
}
fun(mid + 1, right);
}
}
int main()
{

```

```

for(int i = 0; i < n; ++i)
scanf("%d", &A[i]);
fun(0, n-1);
for(int i = 0; i < n; ++i)
    printf("%d ", B[i]);
printf("\n");
}

```

Каков результат работы приведенного кода, если ввести 1 1 1 1 1 0 1 1?

Задача 14. Рассмотрим последовательность из k открывающихся и закрывающихся круглых и фигурных скобок () { }. Среди всех таких последовательностей выделим правильные – те, которые могут быть получены по следующим правилам:

- 1) пустая последовательность правильна;
- 2) если A и B правильны, то и AB правильна;
- 3) если A правильна, то (A) и $\{A\}$ правильны.

Проверьте правильность последовательности, используя число действий (выполняемых операторов присваивания) не превосходящее Ck для некоторой константы C .

Задача 15. На прямой заданы n отрезков $[a_i, b_i]$, $i \in \{1, 2, \dots, n\}$. Предложите алгоритм, находящий наибольшее число k , при котором на прямой найдется точка, покрытая k отрезками. При этом число действий (выполняемых операторов присваивания) должно не превосходить $Cn \log_2 n$ для некоторой константы C .

Вступительный экзамен по дисциплине «Основы информационных технологий» проводится в письменной форме. Абитуриентам предлагается решить 15 задач за 180 минут.

III. Оценка результатов профильного вступительного испытания

Все задачи предполагают наличие решения с пояснениями. Если абитуриент приводит лишь ответ к задаче, то она оценивается в 0 баллов вне зависимости от правильности ответа.

Определение оценки экзаменационной работы осуществляется в два этапа:

1) подсчет количества баллов, фактически набранных абитуриентом, за выполнение каждого задания в соответствии с таблицей 1;

2) перевод суммарного количества баллов в оценку работы абитуриента в соответствии с таблицей 2.

Таблица 1

Максимальное количество баллов за выполнение каждого задания

№ задачи	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Сумма
Балл	1	1	1	2	2	2	2	2	2	3	3	3	3	4	4	35

Таблица 2

Шкала перевода суммарного количества баллов в оценку работы абитуриента

Число баллов	0	1	2	3	4–6	7–12	13–18	19–25	26–30	31–33	34–35
Оценка	0	1	2	3	4	5	6	7	8	9	10

В случае, когда абитуриент присутствовал на вступительном испытании, но экзаменационную работу не выполнял, ему выставляется оценка 0.

Каждое задание может быть:

- выполненным;
- выполненным частично;
- невыполненным.

Задание считается **выполненным**, если ответ абитуриента удовлетворяет следующим критериям:

- правильно выбран способ решения задачи;
- в коде программы допущено не более двух синтаксических ошибок;
- код программы работает во всех требуемых случаях;
- даны точные определения необходимых терминов;
- временная сложность программы удовлетворяет требованиям условия задачи;
- получен правильный ответ.

Задание считается **выполненным частично**, если способ решения задачи выбран правильно, однако в коде программы допущено не менее трех и не более восьми синтаксических ошибок; код программы работает в более чем 75% возможных случаев; временная сложность программы не удовлетворяет требованиям условия задачи.

Задание считается **невыполненным**, если способ решения задачи выбран неправильно; в коде программы допущено более восьми синтаксических ошибок; код программы работает в менее чем 75% возможных случаев.

Оценка за каждое задание должна равняться целому числу баллов.

Если задание выполнено, абитуриент получает максимально возможное число баллов.

Если задание выполнено частично, то абитуриент получает за него не более 75% баллов от максимально возможного числа баллов. Если при этом оценка не является целым числом, то производится округление до ближайшего целого в меньшую сторону.

За невыполненное задание абитуриент получает 0 баллов.

IV. Рекомендуемая литература

Основная

1. *Голицина, О.Л.* Информационные технологии / О.Л. Голицина [и др.]. – М., 2006.
2. *Шифрин, Ю.А.* Информационные технологии / Ю.А. Шифрин. – М., 2002.
3. *Могилев, А.В.* Информатика / А.В. Могилев [и др.]. – М., 2001.
4. *Попов, В.Б.* Основы компьютерных сетей / В.Б. Попов. – М., 2002.
5. *Керниган, Б.* Язык программирования С / Б. Керниган [и др.]. – 2-е изд. – М., 2013.
6. *Кормен, Т.* Алгоритмы. Построение и анализ / Т. Кормен [и др.]. – 3-е изд. – М., 2013.
7. *Олифер, В.Г.* Компьютерные сети. Принципы, технологии, протоколы / В.Г. Олифер [и др.]. – СПб., 2010.
8. *Павловская, Т.А.* С/С++. Структурное и объектно-ориентированное программирование: Практикум / Т.А. Павловская [и др.]. – СПб., 2011.

Дополнительная

1. *Олифер, В.Г.* Основы компьютерных сетей / В.Г. Олифер [и др.]. – СПб., 2009.
2. *Таненбаум, Э.* Архитектура компьютера / Э. Таненбаум, Т. Остин. – 6-е изд. – СПб., 2013.
3. *Поршнеv, С.В.* Компьютерное моделирование физических процессов в пакете MATLAB / С.В. Поршнеv. – М., 2003.