

**ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»**  
**ФИЗИКО-ТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ**  
**Кафедра Компьютерных технологий**



**УТВЕРЖДАЮ:**

ректор по научно-методической  
и учебной работе

Е.И. Скафа

«22» апреля 2020 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**«ОСНОВЫ ПРОГРАММИРОВАНИЯ»**

Направление подготовки: **09.03.01 Информатика и вычислительная техника**

Профиль подготовки: Информатика и вычислительная техника

Образовательная программа: бакалавриат

Квалификация: академический бакалавр

Форма обучения: очная, заочная

Донецк 2020



**УТВЕРЖДАЮ:**

Декан физико-технического факультета

Фоменко С.А.

«17» апреля 2020 г.

М.П.

Программа учебной дисциплины **«Основы программирования»** составлена на основе Государственного образовательного стандарта высшего профессионального образования (ГОС ВПО) по направлению подготовки 09.03.01 Информатика и вычислительная техника, утверждённого приказом Министерства образования и науки ДНР от «21» января 2016 г. №31; «Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики», утверждённого приказом Министерства образования и науки ДНР №1171 от «10» ноября 2017 г.; учебного плана и основной образовательной программы высшего профессионального образования направления подготовки 09.03.01 Информатика и вычислительная техника, разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчик:

*Доцент кафедры компьютерных технологий*

Шарий Т.В.

Программа учебной дисциплины утверждена на заседании кафедры компьютерных технологий

Протокол № 12 от «2» апреля 2020 г.

Зав. кафедрой компьютерных технологий

Ермоленко Т.В.

Программа учебной дисциплины одобрена учебно-методической комиссией физико-технического факультета

Протокол № 5 от «15» апреля 2020 г.

Председатель учебно-методической комиссии физико-технического факультета

Котенко В.Н.

## 1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

Учебная дисциплина «Основы программирования» относится к базовой части профессионального блока и состоит из четырех содержательных модулей: модуль 1 – «Базовые алгоритмические конструкции», модуль 2 – «Пользовательские функции и типы данных», модуль 3 – «Базовые алгоритмы и структуры данных», модуль 4 – «Продвинутые алгоритмы и структуры данных».

Основывается на знаниях школьного курса алгебры, геометрии и информатики. Является основой для изучения следующих дисциплин: «Программирование», «Технологии разработки программного обеспечения».

## 2. СТРУКТУРА ДИСЦИПЛИНЫ

<i>Характеристика учебной дисциплины</i>		
Направление подготовки	09.04.01 Информатика и вычислительная техника	
Магистерская программа	Информатика и вычислительная техника	
Образовательная программа	Бакалавриат	
Квалификация	Академический бакалавр	
Количество содержательных модулей	4	
Дисциплина базовой / вариативной части образовательной программы	Профессиональный блок. Базовая часть	
Формы контроля (МК, экзамен, зачет)	Два модульных контроля, экзамен, зачет	
Показатели	очная форма обучения	заочная форма обучения
Количество зачетных единиц (кредитов)	7	7
Год подготовки	1	1
Семестр	1,2	1,2
Количество часов	252 (126,126)	252 (126,126)
- лекционных	64 (30, 34)	16 (8, 8)
- практических, семинарских	-	-
- лабораторных	64 (30, 34)	16 (8, 8)
- самостоятельной работы	124 (66, 58)	220 (110, 110)
в т.ч. индивидуальное задание		
Недельное количество часов,	8 (8, 8)	8 (8, 8)
в т.ч. аудиторных	4 (4, 4)	1 (1, 1)

## 3. ОПИСАНИЕ ДИСЦИПЛИНЫ

**Цель** – формирование представлений студента о фундаментальных понятиях алгоритмизации, парадигмах и стиле программирования, базовых навыков программирования.

**Задачи** – изучение основ алгоритмизации и программирования; изучение наиболее важных типов данных, структур данных и алгоритмов с умением применять их на практике; формирование информационной культуры студента, в соответствии с современным уровнем и перспективами развития сферы информационных технологий; формирование знаний и навыков в области программирования для дальнейшего обучения и самообучения, научно-исследовательской и профессиональной деятельности.

### **Требования к результатам освоения дисциплины:**

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ГОС ВПО по данному направлению подготовки (профилю):

#### **а) общекультурных (ОК):**

способность работать в коллективе, толерантно воспринимать социальные, этнические, конфессиональные и культурные различия (ОК-6);

способность к самоорганизации и самообразованию (ОК-7).

#### **б) общепрофессиональных (ОПК):**

основательная подготовка по математике для использования математического аппарата при решении прикладных и научных задач в области компьютерной инженерии (ОПК-1);

знание современных методов построения и анализа алгоритмов, основ численных методов и умение их использовать на практике (ОПК-4).

#### **в) профессиональных (ПК):**

##### ***проектно-конструкторская деятельность:***

пользоваться методиками использования программных средств для решения практических задач (ПК-2);

использовать и самостоятельно разрабатывать интерфейсы взаимодействия человека и ЭВМ (ПК-3);

знание принципов программирования, средств современных языков программирования, структур данных (ПК-5).

##### ***Проектно-технологическая деятельность:***

знание особенностей системного программирования, владение методами и средствами разработки элементов системных программ (ПК-10).

##### ***Научно-педагогическая деятельность:***

готовить конспекты лекций, проводить повышение квалификации сотрудников (ПК-17).

##### ***Сервисно-эксплуатационная деятельность:***

инсталлировать, настраивать и сопровождать программное и аппаратное обеспечение информационных и автоматизированных систем (ПК-21).

### **В результате изучения учебной дисциплины студент должен**

#### ***Знать:***

- принципы структурного и модульного программирования;
- этапы решения задач на компьютере;
- типы данных, их возможности и особенности преобразования типов;
- базовые конструкции изучаемого высокоуровневого языка программирования;
- простые и продвинутые алгоритмы поиска и сортировки;
- наиболее используемые фундаментальные структуры данных и абстрактные типы данных.

#### ***Уметь:***

- работать в инструментальной среде разработки;
- кодировать алгоритмы;
- оценивать алгоритмическую сложность решения задач.

#### ***Владеть:***

- навыками составления алгоритмов для решения базовых задач, написания и отладки программ.

#### 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА

Порядковый номер и тема	Краткое содержание темы
	<b><i>Содержательный модуль 1.</i></b> <b>Базовые алгоритмические конструкции</b>
<b>Тема 1.</b> Введение в алгоритмизацию и программирование.	Введение в алгоритмизацию и программирование. Алгоритм, свойства, виды представления. Основные парадигмы современного программирования.
<b>Тема 2.</b> Интегрированная среда разработки MS Visual Studio.	Интегрированная среда разработки MS Visual Studio. Редактор кода, отладчик. Знакомство с .NET.
<b>Тема 3.</b> Линейный вычислительный процесс	Структура программы на языке C#. Простые типы данных C#. Переменные, константы, идентификаторы, операторы, выражения. Работа с консолью.
<b>Тема 4.</b> Разветвляющийся и циклический вычислительный процесс.	Разветвляющийся и циклический вычислительные процессы. Конструкции языка C#: if, switch, for, while, break, continue.
	<b><i>Содержательный модуль 2.</i></b> <b>Пользовательские функции и типы данных</b>
<b>Тема 5.</b> Массивы	Одномерные и двумерные массивы. Свойства массивов. Ступенчатые и прямоугольные массивы. Генерация случайных чисел.
<b>Тема 6.</b> Пользовательские функции. Рекурсия	Пользовательские функции. Передача параметров: модификаторы ref, out. Ключевое слово params. Рекурсивные функции.
<b>Тема 7.</b> Абстрактные синтаксические деревья. РБНФ-выражения.	Синтаксис, семантика, грамматика. Деревья разбора. Абстрактные синтаксические деревья. РБНФ-выражения.
<b>Тема 8.</b> Работа со строками.	Работа со строками. Тип String, StringBuilder. Введение в регулярные выражения. Основные элементы синтаксиса.
<b>Тема 9.</b> Перечисляемый тип данных. Структуры.	Перечисляемый тип данных enum. Структуры struct. Работа с датой и временем (DateTime, TimeSpan).
	<b><i>Содержательный модуль 3.</i></b> <b>Базовые алгоритмы и структуры данных</b>
<b>Тема 10.</b> Работа с файлами	Работа с двоичными и текстовыми файлами. Тип FileStream, BinaryReader, BinaryWriter, StreamReader, StreamWriter.
<b>Тема 11.</b> Простые алгоритмы сортировки	Простые алгоритмы сортировки: выбором, вставками, пузырьком, шейкерная, Шелла. Введение в анализ сложности алгоритмов.
<b>Тема 12.</b> Простые алгоритмы поиска	Алгоритмы поиска: линейный поиск перебором, бинарный, интерполяционный поиск.
<b>Тема 13.</b> Поиск последовательностей	Поиск последовательностей: перебор, алгоритм Кнута-Мориса-Пракса, алгоритм Бойера-Мура-Хорспула.
<b>Тема 14.</b> Фундаментальные структуры данных и абстрактные типы данных	Фундаментальные структуры данных (ФСД) и абстрактные типы данных (АТД). Односвязные и двусвязные списки. Циклические списки.

<b>Тема 15.</b> Стек, очередь, дек	АТД стека, очереди, дека
<b>Тема 16.</b> Деревья	Деревья. Обход деревьев. Деревья бинарного поиска. Сбалансированные деревья. В-деревья.
	<b>Содержательный модуль 4.</b> <b>Продвинутое алгоритмы и структуры данных</b>
<b>Тема 17.</b> Хеш-таблицы	Хеш-таблицы. Хеш-функции
<b>Тема 18.</b> Коллекции в .NET	Коллекции в .NET: List, LinkedList, Dictionary, Stack, Queue
<b>Тема 19.</b> Двоичная куча. Очередь с приоритетами	Двоичная куча. АТД очереди с приоритетами. Пирамидальная сортировка
<b>Тема 20.</b> Продвинутое алгоритмы сортировки	Стратегия «Разделяй и властвуй». Быстрая сортировка. Сортировка слиянием. Сортировка подсчетами
<b>Тема 21.</b> Графы	Графы. Представление графа в памяти. Поиск в глубину и ширину. Поиск кратчайших путей в графе. Минимальные остовные деревья.
<b>Тема 22.</b> Динамическое программирование	Принципы динамического программирования. Задача о рюкзаке.

Курс дисциплины «Основы программирования» предусматривает следующие **формы организации учебного процесса**:

1. лекции;
2. лабораторные занятия;
3. самостоятельная работа студента.

По источнику передачи и восприятия учебной информации используются словесные (лекция, беседа), наглядные (слайды, иллюстрации, коды программ), практические (исследования, упражнения, лабораторные работы) методы.

По характеру познавательной деятельности студентов используются объяснительно-иллюстративные и репродуктивные методы, проблемное преподавание, частично-поисковый и исследовательский методы.

В зависимости от основной дидактической цели и задач используются методы устного изложения знаний, закрепление учебного материала, самостоятельной работы студентов по осмыслению и усвоению нового материала, работы по применению знаний на практике и выработке умений и навыков, проверки и оценки знаний, умений и навыков.

Используются следующие методы контроля:

1. устный контроль (экспресс-опрос на лекциях);
2. проверка конспектов;
3. защита лабораторных работ;
4. проверка самостоятельных работ;
5. модульные контрольные работы;
6. итоговый контроль (экзаменационные билеты).

## Тематический план

Содержательный модуль 1												
Названия содержательных модулей и тем	Количество часов											
	Очная форма обучения						Заочная форма обучения					
	всего	в т.ч.					всего	в т.ч.				
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа
<b>Тема 1.</b> Введение в алгоритмизацию и программирование.	8	2		2	4		14	1		1	12	
<b>Тема 2.</b> Интегрированная среда разработки MS Visual Studio.	16	2		2	12		14	1		1	12	
<b>Тема 3.</b> Линейный вычислительный процесс	16	4		4	8		16	1		1	14	
<b>Тема 4.</b> Разветвляющийся и циклический вычислительный процесс.	22	4		4	14		18	1		1	16	
<b>Итого по 1-му содержательному модулю</b>	62	12		12	38		62	4		4	54	

Содержательный модуль 2												
Названия содержательных модулей и тем	Количество часов											
	Очная форма обучения						Заочная форма обучения					
	всего	в т.ч.					всего	в т.ч.				
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа
<b>Тема 5.</b> Массивы	10	2		4	4		12	1		1	10	
<b>Тема 6.</b> Пользовательские функции. Рекурсия	14	4		2	8		12	1		1	10	
<b>Тема 7.</b> Абстрактные синтаксические деревья. РБНФ-выражения.	6	2			4		12				12	
<b>Тема 8.</b> Работа со строками.	20	6		6	8		14	1		1	12	
<b>Тема 9.</b> Перечисляемый тип данных. Структуры	14	4		6	4		14	1		1	12	
<b>Итого по 2-му содержательному модулю</b>	64	18		18	28		64	4		4	56	



Содержательный модуль 3												
Названия содержательных модулей и тем	Количество часов											
	Очная форма обучения						Заочная форма обучения					
	всего	в т.ч.					всего	в т.ч.				
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа
<b>Тема 10.</b> Работа с файлами	10	2		4	4		10	1		1	8	
<b>Тема 11.</b> Простые алгоритмы сортировки	6	2		2	2		6	1		1	4	
<b>Тема 12.</b> Простые алгоритмы поиска	6	2		2	2		6	1		1	4	
<b>Тема 13.</b> Поиск последовательностей	8	2		2	4		8				8	
<b>Тема 14.</b> Фундаментальные структуры данных и абстрактные типы данных	12	4		2	6		10	1		1	8	
<b>Тема 15.</b> Стек, очередь, дек	8	2		2	4		10	1		1	8	
<b>Тема 16.</b> Деревья	12	4		4	4		12	1		1	10	
<b>Итого по 3-му содержательному модулю</b>	62	18		18	26		62	6		6	50	

Содержательный модуль 4												
Названия содержательных модулей и тем	Количество часов											
	Очная форма обучения						Заочная форма обучения					
	всего	в т.ч.					всего	в т.ч.				
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа
<b>Тема 17.</b> Хеш-таблицы	8	2		2	4		8				8	
<b>Тема 18.</b> Коллекции в .NET	12	2		6	4		14	1		1	12	
<b>Тема 19.</b> Двоичная куча. Очередь с приоритетами	8	2		2	4		8				8	
<b>Тема 20.</b> Продвинутое алгоритмы сортировки	10	4		2	4		10	1		1	8	
<b>Тема 21.</b> Графы	16	4		4	8		12				12	
<b>Тема 22.</b> Динамическое программирование	10	2			8		12				12	
<b>Итого по 4-му содержательному модулю</b>	64	16		16	32		64	2		2	60	
<b>Всего часов</b>	252	64		64	124		252	16		16	220	

## 5. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРОВЕДЕНИЯ ЛЕКЦИОННЫХ, ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

### Темы лекционных занятий

<b>№ п/п</b>	<b>Название темы</b>	<b>Количество часов</b>
1.	Введение в алгоритмизацию и программирование.	2
2.	Интегрированная среда разработки MS Visual Studio.	2
3.	Линейный вычислительный процесс	4
4.	Разветвляющийся и циклический вычислительный процесс.	4
5.	Массивы	2
6.	Пользовательские функции. Рекурсия	4
7.	Абстрактные синтаксические деревья. РБНФ-выражения.	2
8.	Работа со строками.	6
9.	Перечисляемый тип данных. Структуры.	4
10.	Работа с файлами	2
11.	Простые алгоритмы сортировки	2
12.	Простые алгоритмы поиска	2
13.	Поиск последовательностей	2
14.	Фундаментальные структуры данных и абстрактные типы данных	4
15.	Стек, очередь, дек	2
16.	Деревья	4
17.	Хеш-таблицы	2
18.	Коллекции в .NET	2
19.	Двоичная куча. Очередь с приоритетами	2
20.	Продвинутое алгоритмы сортировки	4
21.	Графы	4
22.	Динамическое программирование	2
	<b>ВСЕГО</b>	<b>64</b>

### Темы лабораторных занятий

<b>№ п/п</b>	<b>Название темы</b>	<b>Количество часов</b>
1.	Базовые типы данных языка C#. Работа с консолью	6
2.	Программирование ветвления и циклов на языке C#	6
3.	Массивы. Пользовательские функции	6
4.	Работа со строками в C#. Знакомство с регулярными выражениями	6
5.	Перечисления и структуры в C#	6
6.	Работа с файлами	8
7.	Простые алгоритмы сортировки. Анализ алгоритмов	8
8.	Алгоритмы поиска	6
9.	Абстрактные типы данных. Коллекции в .NET	8
10.	Продвинутое алгоритмы сортировки. Алгоритмы на графах	4
	<b>ВСЕГО</b>	<b>64</b>

## 6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная работа студентов по курсу «Основы программирования» предусматривает изучение дополнительной технической литературы и интернет-источников, рекомендуемых этой программой; самостоятельную разработку алгоритмов и текстов программ лабораторных работ, изучение дополнительного инструментария. При желании студент может подготовить реферат или доклад по одной из приведенных ниже тем:

1. Основные парадигмы современного программирования. Сравнительный анализ с примерами кода.
2. Интегрированная среда разработки MS Visual Studio Community. Дополнительные инструменты. Ускорение процесса разработки программного обеспечения.
3. Компиляция. Линковка. Сравнение инструментов.
4. Рекурсия. Примеры кодов рекурсивных алгоритмов. Фракталы.
5. Абстрактные синтаксические деревья. РБНФ-выражения.
6. Регулярные выражения. Примеры задач с решениями на основе строковых методов и регулярных выражений.
7. Работа с файлами. Разбор популярных форматов файлов (wav, mp3, bmp и др.).
8. Деревья бинарного поиска. Сбалансированные деревья. В-деревья.
9. Хеш-таблицы. Хеш-функции. Использование в современных фреймворках.
10. Коллекции в .NET с нюансами реализации.
11. Поразрядная сортировка. Экзотические сортировки.
12. Графы. Продвинутое алгоритмы.
13. Динамическое программирование. Примеры задач с решением.

<b>№ п/п</b>	<b><i>Название темы</i></b>	<b><i>Количество часов</i></b>
1.	Введение в алгоритмизацию и программирование.	4
2.	Интегрированная среда разработки MS Visual Studio.	12
3.	Линейный вычислительный процесс	8
4.	Разветвляющийся и циклический вычислительный процесс.	14
5.	Массивы	4
6.	Пользовательские функции. Рекурсия	8
7.	Абстрактные синтаксические деревья. РБНФ-выражения.	4
8.	Работа со строками.	8
9.	Перечисляемый тип данных. Структуры.	4
10.	Работа с файлами	4
11.	Простые алгоритмы сортировки	2
12.	Простые алгоритмы поиска	2
13.	Поиск последовательностей	4
14.	Фундаментальные структуры данных и абстрактные типы данных	6
15.	Стек, очередь, дек	4
16.	Деревья	4
17.	Хеш-таблицы	4
18.	Коллекции в .NET	4
19.	Двоичная куча. Очередь с приоритетами	4
20.	Продвинутое алгоритмы сортировки	4

21.	Графы	8
22.	Динамическое программирование	8
	<b>ВСЕГО</b>	<b>124</b>

## 7. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Индивидуальные задания предусмотрены к каждой лабораторной работе. Полный перечень всех заданий приведен в УМКД дисциплины.

### Образец индивидуального задания

Написать программу в соответствии с вариантом (из лабораторной работы № 2).

#### Вариант 1.

1. Ввести с экрана координаты точки (2 вещественных числа). Если хотя бы одна из координат равна 0, вывести на экран число 0. Иначе – вывести на экран номер четверти, в которой находится точка.
2. Ввести число от 1 до 10000. Вывести на экран все четные делители числа.

#### Вариант 2.

1. Ввести 2 вещественных числа. Если хотя бы одно из них меньше 0, вывести сообщение «Еггг». Иначе – вывести на экран квадратный корень меньшего числа.
2. Ввести число от 1 до 10000. Вывести на экран все нечетные делители числа.

#### Вариант 3.

1. Ввести с экрана координаты точки А (2 вещественных числа) и точки В (2 вещественных числа). Если точки находятся в одной четверти, вывести на экран true, иначе – false. Если какие-либо координаты точек равны 0, выдать строку «Zero coord».
2. Ввести число от 1 до 100000. Вывести на экран true, если число является степенью тройки, иначе – вывести false (использовать операцию логарифмирования нельзя).

...

## 8. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Алгоритм. Свойства алгоритма. Виды представления алгоритма.
2. Основные парадигмы и подходы современного программирования.
3. Интегрированная среда разработки MS Visual Studio. Работа с проектами. Отладка программ. Конфигурации Debug и Release.
4. Основные элементы фреймворка .NET: общезыковая среда выполнения (CLR) и базовая библиотека классов (BCL).
5. Процесс создания программы. Компиляция и линковка. Пункт меню Build и Debug в MS Visual Studio.
6. Простые типы данных .NET. Преобразование и приведение простых типов. Типизация и ключевое слово var.
7. Арифметические операции. Постфиксный и префиксный инкремент (декремент). Математические операции (возведение в степень, тригонометрия, логарифмы).
8. Простые и составные логические выражения. Логические операторы. Особенности "укороченных" логических операторов. Таблицы истинности.
9. Ветвления. Операторы if, switch, goto. Тернарный оператор (?).
10. Циклы. Операторы for, while, do while, continue, break.

11. Массивы. Оператор foreach. Виды двумерных массивов. Инициализация одномерных и двумерных массивов. Генерация случайных чисел.
12. Пользовательские функции. Ключевые слова params, ref, out. Параметры по умолчанию. Оператор return.
13. Рекурсия. Базовый случай, шаг рекурсии. Стек вызовов. Рекурсивные алгоритмы нахождения НОД двух чисел и факториала числа.
14. Работа со строками как с массивом символов. Тип string. Основные операции со строками (поиск, вставка, удаление, замена подстроки).
15. Регулярные выражения. Основные синтаксические элементы регулярных выражений. Квантификаторы.
16. Тип данных - перечисление. Синтаксис. Занимаемый объем памяти. Модификации. Примеры.
17. Структуры в C#. Работа с датой и временем в C#.
18. Структура программы. Переменные и константы. Идентификаторы. Операторы. Выражения.
19. Абстрактные синтаксические деревья. РБНФ-выражения. Примеры.

## 9.ОБРАЗЕЦ МОДУЛЬНОГО КОНТРОЛЯ

ГОУ ВПО «Донецкий национальный университет»  
Физико-технический факультет

Направление подготовки	09.03.01 - Информатика и вычислительная техника
Профиль	Информатика и вычислительная техника
Программа подготовки	бакалавриат
Семестр	1
Учебная дисциплина	Основы программирования

### МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА №1 ВАРИАНТ №1

1. Ветвления. Операторы if, switch, goto. Тернарный оператор (?).
2. Дано число N. Найти ближайшее к нему число, являющееся степенью двойки (2, 4, 8, 16, 32, 64 и т.д.).
3. Написать программу сложения первого отрицательного и последнего положительного элементов в массиве.
4. Что будет выведено на консоль в результате выполнения кода:

```

int x = 105;
int y = x / 100;
if (y > 1)
    y = ++x + 10;
else
    y = ++x;
Console.WriteLine("{0} and {1}", x, y + 1);

```

Утверждено на заседании кафедры компьютерных технологий,  
протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой  
Преподаватель

\_\_\_\_\_  
\_\_\_\_\_

### Критерии оценивания модульного контроля

<i>Номер задания</i>	<i>Количество баллов</i>
1	5
2	4
3	4
4	2
<b>Всего</b>	<b>15</b>

ГОУ ВПО «Донецкий национальный университет»  
Физико-технический факультет

Направление подготовки 09.03.01 - Информатика и вычислительная техника  
Профиль Информатика и вычислительная техника  
Программа подготовки бакалавриат  
Семестр 2  
Учебная дисциплина Основы программирования

### МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА №2 ВАРИАНТ №1

1. Дан массив:  
2 3 10 5 7 6 4 12 20 9 11 22 24 25  
Написать код и трассировку первых 10 шагов (обменов) алгоритма сортировки массива вставками.
2. Написать код и трассировку алгоритма поиска числа 20 методом бинарного поиска.
3. Написать код на основе Queue<double> для получения такой коллекции данных, что при извлечении из нее данных мы получим последовательно числа 15.5, 2.0, 9.4, 4.1, 7.

Утверждено на заседании кафедры компьютерных технологий,  
протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой  
Преподаватель

\_\_\_\_\_  
\_\_\_\_\_

### Критерии оценивания модульного контроля

<i>Номер задания</i>	<i>Количество баллов</i>
1	10
2	10
3	5
<b>Всего</b>	<b>25</b>

## 10. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

### *Теоретические вопросы к экзамену*

1. Алгоритм. Свойства алгоритма. Виды представления алгоритма.
2. Основные парадигмы и подходы современного программирования.
3. Интегрированная среда разработки MS Visual Studio. Работа с проектами. Отладка программ. Конфигурации Debug и Release.
4. Основные элементы фреймворка .NET: общезыковая среда выполнения (CLR) и базовая библиотека классов (BCL).
5. Процесс создания программы. Компиляция и линковка. Пункт меню Build и Debug в MS Visual Studio.
6. Простые типы данных .NET. Преобразование и приведение простых типов. Типизация и ключевое слово var.
7. Арифметические операции. Постфиксный и префиксный инкремент (декремент). Математические операции (возведение в степень, тригонометрия, логарифмы).
8. Простые и составные логические выражения. Логические операторы. Особенности "укороченных" логических операторов. Таблицы истинности.
9. Ветвления. Операторы if, switch, goto. Тернарный оператор (?).
10. Циклы. Операторы for, while, do while, continue, break.
11. Массивы. Оператор foreach. Виды двумерных массивов. Инициализация одномерных и двумерных массивов. Генерация случайных чисел.
12. Пользовательские функции. Ключевые слова params, ref, out. Параметры по умолчанию. Оператор return.
13. Рекурсия. Базовый случай, шаг рекурсии. Стек вызовов. Рекурсивные алгоритмы нахождения НОД двух чисел и факториала числа.
14. Работа со строками как с массивом символов. Тип string. Основные операции со строками (поиск, вставка, удаление, замена подстроки).
15. Регулярные выражения. Основные синтаксические элементы регулярных выражений. Квантификаторы.
16. Тип данных - перечисление. Синтаксис. Занимаемый объем памяти. Модификации. Примеры.
17. Структуры в C#. Работа с датой и временем в C#.
18. Структура программы. Переменные и константы. Идентификаторы. Операторы. Выражения.
19. Абстрактные синтаксические деревья. РБНФ-выражения. Примеры.
20. Основы анализа сложности алгоритмов. O-нотация.



ГОУ ВПО «Донецкий национальный университет»  
Физико-технический факультет

Направление подготовки 09.03.01 - Информатика и вычислительная техника  
Профиль Информатика и вычислительная техника  
Программа подготовки бакалавриат  
Семестр 1  
Учебная дисциплина Основы программирования

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 8

1. Простые и составные логические выражения. Логические операторы. Особенности "укороченных" логических операторов. Таблицы истинности.
2. Даны два массива по 8 элементов. Сформировать и вывести на консоль третий массив, в котором содержатся только те элементы из первого массива, которых нет во втором.
3. Что будет выведено на консоль в результате выполнения кода:  

```
int x = 15; double y = x / 10.0; int res = 0;
if (y > 1) res = 1;
Console.WriteLine("res={0}", res);
```

Утверждено на заседании кафедры компьютерных технологий,  
протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой \_\_\_\_\_  
Преподаватель \_\_\_\_\_

#### Критерии оценивания экзамена

<i>Номер задания</i>	<i>Количество баллов</i>
1	25
2	20
3	5
<b>Всего</b>	<b>50</b>

## 11. ОБРАЗЕЦ ТЕСТОВОГО ЗАДАНИЯ

Тестовые задания не предусмотрены.

## 12. КРИТЕРИИ ОЦЕНИВАНИЯ

По курсу предполагается проведение промежуточной аттестации в виде 2 модульных контролей, экзамена, зачета, выполнение лабораторных работ.

*Распределение баллов, которые могут получить студенты  
в процессе изучения дисциплины*

**Семестр 1**

	Содержательный модуль №1				Содержательный модуль №2				Экзамен	Всего
	Лабораторные работы		Мод. контр. работа	Всего С.М. №1	Лабораторные работы			Всего С.М. №2		
	№1	№2			№3	№4	№5			
Макс. балл	7	7	15	29	7	7	7	21	50	100

**Семестр 2**

	Содержательный модуль №1				Содержательный модуль №2			Всего	
	Лабораторные работы		Мод. контр. работа	Всего С.М. №1	Лабораторные работы		Всего С.М. №2		
	№1	№2			№3	№4			№5
Макс. балл	15	15	25	55	15	15	15	45	100

***Шкала соответствия баллов национальной шкале***

Оценка по шкале ECTS	Оценка по 100-балльной шкале	Оценка по государственной шкале (экзамен, дифференцированный зачет)	Оценка по государственной шкале (зачет)
<b>A</b>	90-100	5 (отлично)	зачтено
<b>B</b>	80-89	4 (хорошо)	зачтено
<b>C</b>	75-79	4 (хорошо)	зачтено
<b>D</b>	70-74	3 (удовлетворительно)	зачтено
<b>E</b>	60-69	3 (удовлетворительно)	зачтено
<b>FX</b>	35-59	2 (неудовлетворительно) с возможностью повторной сдачи	не зачтено
<b>F</b>	0-34	2 (неудовлетворительно) с возможностью повторной сдачи при условии обязательного набора дополнительных баллов	не зачтено

Оценка за овладение курса выставляется по следующим критериям:

– Оценку «отлично» заслуживает студент, который обнаружил глубокие знания при ответах на теоретические вопросы по темам курса, а также выполнил практические задания в полном объеме и набрал более 90 баллов.

– Оценку «хорошо» заслуживает студент, сделавший ошибки в теоретических или практических ответах, которые могут быть интерпретированы как малозначительные для вопросов, которые рассматривались. Студент должен набрать более 75 баллов.

– Оценку «удовлетворительно» заслуживает студент, который выполнил задания неполно и с ошибками, но при этом набрал более 60 баллов.

– Оценку «неудовлетворительно» заслуживает студент, который не выполнил большинства теоретических и практических задач и набрал менее 60 баллов.

## 12. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Для проведения лекционных занятий требуется аудитория на поток, оборудованная мультимедийным проектором и экраном, или интерактивной доской. Для проведения лабораторных занятий по дисциплине необходим оборудованный ПЭВМ или ноутбуками компьютерный класс с возможностью выхода в Интернет.

## 13. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№ п/п	Наименование	Кол-во экземпляров в библиотеке ДонНУ	Наличие электронной версии в ЭБС
<i>Основная литература</i>			
1.	Nakov S. Fundamentals of Computer Programming with C# [Текст] / S. Nakov & Co. – BASD, Sofia, 2013. – 1121 p.	-	Да
<i>Дополнительная литература</i>			
2.	Стивенс Р. Алгоритмы. Теория и практическое применение [Текст] / Р. Стивенс. – М.: Издательство «Э», 2016. – 544 с.	-	Нет
3.	Дасгупта С. Алгоритмы [Текст] / С. Дасгупта, Х. Пападимитриу, У. Вазири. – М.: МЦНМО, 2014. – 320 с.	-	Нет
4.	Златопольский Д. М. Сборник задач по программированию [Текст] / Д. М. Златопольский. – СПб.: БХВ-Петербург, 2007. – 240 с.	-	Нет
5.	Нейгел К. C#5.0 и платформа .NET 4.5 для профессионалов [Текст] / К. Нейгел, Б. Ивсен., Дж. Глини. – М.: Издательский дом «Вильямс», 2014. – 1440 с.	1	Нет
6.	Кормен Т. Алгоритмы: Построение и анализ [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М.: МЦНМО, 1999. – 960 с.	1	Нет

## 14. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. Реализации основных алгоритмов на языке C# (автор – Шарий Т.В.). URL: <https://github.com/ar1st0crat/AlgorithmsCheatSheet> (дата обращения 18.03.2020 г.).
2. C#. Уроки программирования с нуля. URL: <http://mycsharp.ru> (дата обращения 18.03.2020 г.).
3. Полное руководство по языку программирования C#. URL: <http://metanit.com/sharp/tutorial> (дата обращения 18.03.2020 г.).

## 15. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. IDE Visual Studio Community любой версии (рекомендованы версии от 2015 г.).
2. IDE MonoDevelop (для пользователей ОС Linux).
3. Текстовый редактор Notepad++ с дополнением HexEditor.

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2020 год.  
Протокол № 12 от «2» апреля 2020 г.  
Заведующий кафедрой Ермоленко Т.В.

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2021 год.  
Протокол № \_\_\_\_ от «\_\_\_\_» \_\_\_\_\_ 2021 г.  
Заведующий кафедрой

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2022 год.  
Протокол № \_\_\_\_ от «\_\_\_\_» \_\_\_\_\_ 2022 г.  
Заведующий кафедрой

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2023 год.  
Протокол № \_\_\_\_ от «\_\_\_\_» \_\_\_\_\_ 2023 г.  
Заведующий кафедрой