

## Содержание первого семестра дисциплины «Информатика (C#)»:

Неделя	Темы занятий
1, 2	<p>1.1 Введение в алгоритмизацию</p> <ul style="list-style-type: none"> <li>▪ Понятие алгоритма и компьютерной программы</li> <li>▪ Разрешимые и неразрешимые задачи, алгоритмическая сложность</li> <li>▪ Структура компьютера по фон Нейману</li> <li>▪ Языки низкого и высокого уровня</li> <li>▪ Компилируемые и интерпретируемые языки (примеры, различия)</li> </ul> <p>1.2 Технологии Microsoft .NET NET Framework. История, идеология</p> <ul style="list-style-type: none"> <li>▪ CLR, CTS, CLS, MSIL</li> <li>▪ Языки .NET. Межъязыковое взаимодействие</li> <li>▪ Язык C#</li> <li>▪ Этапы компиляции и исполнения программ в .NET Framework на примере языка C#</li> </ul>
1	<p>1. Структурирование блок-схем. Введение в версионный контроль</p> <p>2. Знакомство с C#. Компиляция и исполнение программ</p>
2	<p>3. Примитивные типы, операции</p> <p>4. Простые программы</p>
3, 4	<p>2. Типы данных, управляющие операторы языка</p> <ul style="list-style-type: none"> <li>▪ Основы типов. Ссылочные типы и типы значения</li> <li>▪ Операции (арифметические, логические и т.д.)</li> <li>▪ Обзор условных и циклических операторов языка C#</li> <li>▪ Аргументы командной строки</li> <li>▪ Консольный ввод/вывод</li> </ul>
3	<p>5. Ввод/вывод данных</p> <p>6. Условный оператор</p>
4	<p>7. Циклы</p> <p>8. Принадлежность точки фигуре</p>
5, 6	<p>3. Базовые алгоритмы обработки данных</p> <ul style="list-style-type: none"> <li>▪ Понятие подпрограммы</li> <li>▪ Методы в языке C#</li> <li>▪ Способы передачи параметров в методы</li> <li>▪ Подходы к построению алгоритмов</li> <li>▪ Нисходящее (bottom-up) и восходящее (top-down) проектирование алгоритмов</li> <li>▪ Рекуррентные вычисления</li> <li>▪ Рекурсия</li> </ul>
5	<p>9. НОД, НОК, Возведение в степень</p> <p>10. Схема Горнера, суммы (конечные, бесконечные)</p>
6	<p>11. Поточковая обработка данных (кванторы)</p> <p>12. Рекурсия</p>
7, 8	<p>4. Массивы и действия с ними</p> <ul style="list-style-type: none"> <li>▪ Массивы и хранение данных в памяти</li> <li>▪ Одномерные массивы</li> <li>▪ Многомерные массивы</li> <li>▪ Ступенчатые массивы</li> <li>▪ Алгоритмы обработки массивов</li> <li>▪ Сортировка, поиск, агрегатные функции</li> </ul>
7	<p>13. Одномерные массивы (поиск, сортировка, агрегатные функции)</p> <p>14. Двумерные массивы</p>
8	<p>15. Ступенчатые массивы</p> <p>16. Задачи на матрицы</p>
9, 10	<p>5. Символы и строки</p> <ul style="list-style-type: none"> <li>▪ Тип char, символы и номера ASCII</li> <li>▪ Класс String</li> <li>▪ Ввод/вывод строк</li> <li>▪ StringBuilder, StringWriter</li> <li>▪ Свойства Length и Chars</li> <li>▪ Операции на строках</li> </ul>
9	<p>17. Решение задач на работу со строками (ввод/вывод)</p> <p>18. Решение задач на работу со строками (сравнение строк)</p>
10	<p>19. Решение задач на работу со строками (поиск символа, поиск подстроки)</p>

Неделя	Темы занятий
11, 12	7. Основы объектно-ориентированного программирования ч.1. <ul style="list-style-type: none"> <li>▪ Принципы ООП.</li> <li>▪ Классы и объекты.</li> <li>▪ Структуры и классы в .NET</li> <li>▪ Методы, поля, модификаторы доступа, свойства, индексы</li> <li>▪ Вложенные классы</li> <li>▪ Взаимосвязь между классами (зацепление – coupling)</li> <li>▪ Композиция и наследование</li> </ul>
11	20. Структуры, классы, конструкторы, деструкторы 21. Инкапсуляция. Методы. «get», «set» методы, свойства. Вложенные типы.
12	22. Реализация объектов из различных предметных областей. Структуры и классы. 23. Реализация объектов из различных предметных областей. Композиция.
13, 14	8. Основы объектно-ориентированного программирования ч.2. <ul style="list-style-type: none"> <li>▪ Наследование. Модификаторы доступа, конструкторы и наследование, восходящее и нисходящее (up/downcast), боксинг/анбоксинг</li> <li>▪ Полиморфизм. Перегрузка и переопределение. Абстрактные классы. «Запечатанные» классы.</li> </ul>
13	24. Наследование. Контроль, модификаторы, проектирование иерархий. Полиморфизм. Перегрузка и переопределение методов. 25. Наследование. Контроль, модификаторы, проектирование иерархий. Полиморфизм. Перегрузка и переопределение методов.
14	26. Реализация объектов из различных предметных областей. Особенности наследования. Преобразования типов. 27. Реализация объектов из различных предметных областей. Особенности наследования. Преобразования типов.
15, 16	9. Интерфейсы. Введение в принципы проектирования. Роль в тестировании. Роль в гибкости и расширяемости. Полиморфизм. SOLID принципы, идеи IoC, DI. Проектирование. Принципы проектирования. Шаблоны и архитектурные шаблоны.
15	28. Реализация объектов из различных предметных областей. Принципы проектирования. CQRS, DbC 29. Реализация объектов из различных предметных областей. Принципы проектирования. Примеры IoC, DI
16	30. Реализация объектов из различных предметных областей. Принципы проектирования. SOLID 31. Реализация объектов из различных предметных областей. Принципы проектирования. Архитектурные шаблоны (MVC, MVP, MVVM).
17	10. Разбор полетов.
17	32. Обзор проектов, обсуждение результатов семестра.
Quiz 2	
FINAL EXAM	

## Распределение баллов:

По результатам семестра можно набрать максимум 100 баллов, которые распределяются следующим образом:

<b>Текущий контроль:</b>		<b>Итоговый контроль:</b>	
Летучки x 10шт.	50 баллов	Коллоквиумы x 2 шт.	50 баллов
Контрольные x 2 шт.	10 баллов	Итоговый экзамен x 1шт.	30 баллов
Домашние задания x 5 шт.	10 баллов		20 баллов
	30 баллов		

**Летучки** представляют собой задания на решение небольших по объему задач в ограниченное время (5-10 мин.). Летучки выполняются письменно во время практических занятий в аудитории.

**Контрольные работы** представляют собой задания на практическое выполнение задач среднего объема в ограниченное время (40 – 60 минут). Контрольные выполняются на компьютерах во время практических занятий в аудитории.

**Домашние задания** представляют собой наборы задач (problem set) на самостоятельное выполнение во внеурочное время. Прием решений ограничивается сроками мягкого и жесткого дедлайна. Решение, сданное в рамках мягкого дедлайна оценивается в рамках максимально возможных баллов. Решение, сданное после мягкого дедлайна, но до жесткого дедлайна оценивается в половину возможных баллов (штраф 50%). Решения, сданные после жесткого дедлайна, не оцениваются.

**Коллоквиум** представляет собой тест-викторину (quiz) с письменным ответом на вопросы в ограниченное время (40 – 80 минут), выполняемый во время практических занятий в аудитории. Вопросы могут быть представлены в виде теста с одним/несколькими правильными ответами, в виде вопроса на вычисление ответа (строка, число и т.д.) и в виде вопроса с необходимостью развернутого ответа (описание определения, алгоритма, небольшой программы).

**Итоговый экзамен** аналогичен по форме коллоквиуму. Отличие экзамена от коллоквиума в более широком охвате материала (коллоквиум ограничен одним разделом семестра, экзамен – обоими разделами семестра) и объеме. Время, отведенное на решение теста-викторины итогового экзамена, составляет 80-120 минут. Итоговый экзамен проходит во время сессии в отдельно выделенный день.

## Сроки сдачи домашних заданий и критерии оценки

Оценка за домашнее задание вычисляется по формуле:  $intime \times scope (correctness + design\&style)$ , где

$intime$  – штрафной коэффициент за несвоевременную сдачу

(1 – вовремя, 0.5 до жесткого дедлайна, 0 – после жесткого дедлайна)

$scope$  – мера того, насколько решение соответствует требованиям задания.

Оценивается от 0 до 10.

$correctness$  – мера того, насколько программный код корректен (не содержит ошибок и багов).

Оценивается от 0 до 10.

$design\&style$  – мера того, насколько программа хорошо спроектирована и удобна для использования.

Измеряется от 0 до 10.

Итоговая оценка за домашние задания складывается из нормализованной (приведённой к 30 баллам) суммарной оценки за 4 лучшие домашние работы. Таким образом, один худший результат отбрасывается.

### Темы и сроки выполнения домашних работ:

**Домашняя работа 1** (2-3 неделя). Тема: «Примитивные типы и операции, ввод/вывод, условия, циклы»

Мягкий дедлайн – конец 3-ей недели. Жесткий дедлайн – конец 4-ой недели.

**Домашняя работа 2** (5-6 неделя). Тема: «Вычисление конечных и бесконечных сумм»

Мягкий дедлайн – конец 6-ой недели. Жесткий дедлайн – конец 7-ой недели.

**Домашняя работа 3** (7-8 неделя). Тема: «Массивы (1D, 2D), методы»

Мягкий дедлайн – конец 8-ой недели. Жесткий дедлайн – конец 9-ой недели.

**Домашняя работа 4** (12-13 неделя). Тема: «Массивы объектов»

Мягкий дедлайн – конец 13-ой недели. Жесткий дедлайн – конец 14-ой недели.

**Домашняя работа 5** (15-16 неделя). Тема: «ООП. Проектирование классов и иерархий»

Мягкий дедлайн – конец 16-ой недели. Жесткий дедлайн – конец 17-ой недели.