

Автономная некоммерческая организация
«Красноярский детский технопарк «Кванториум»

РЕКОМЕНДОВАНО
методическим советом

Протокол № 11
от «30» мая 2024 г.

УТВЕРЖДАЮ

Генеральный директор
Кенниг С.Р.

Приказ № 11
от «30» мая 2024 г.



Дополнительная общеобразовательная общеразвивающая программа
технической направленности

«Основы программирования»

Срок реализации:

1 год

Возраст детей:

12-18 лет

Составитель программы:

Цыганов М.В.

г. Красноярск, 2024 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования» (далее – программа), технической направленности, нацелена на изучение языка программирования С++, так как он является одним из наиболее востребованных на рынке труда. Программа на базовом уровне освещает практические основы программирования. В ходе обучения предстоит решить множество задач, охватывающих основные базовые конструкции языка С++. Такой опыт будет полезен всем, кто хочет углубиться в изучение программирования.

Данная программа формирует профессиональные компетенции, которые позволят обучающимся в будущем создавать собственные программы, игры, сайты. Программа направлена на развитие у детей логического мышления, вычислительного мышления, совершенствование первичных навыков программирования, представления о профессии программиста и специалиста в области информационных технологий.

1.1 Актуальность программы

Данная программа нацелена на обучающихся, не имеющих навыков программирования, позволяет им увидеть всю многогранность компьютерных наук.

Изучение основ программирования способствуют формированию современного научного мировоззрения, развитию интеллектуальных способностей и познавательных интересов обучающихся, развивает способности к самообразованию, умению ставить задачи, планировать их выполнение, умению разбивать решение задачи на подзадачи, способность работать с информацией: находить, оценивать и использовать информацию из различных источников. Программирование — это не просто знание какого-то языка или технологии, это способность сформулировать проблему, найти возможные решения этой проблемы путём определения алгоритмов её решения. После прохождения данной программы ребёнку будет проще определить наиболее интересующую его область ИТ и в дальнейшем уже осознанно подходить к выбору дальнейшего направления обучения.

1.2 Отличительные особенности

Данная программа разработана в соответствии с нормативными правовыми актами в области образования:

Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

Указом Президента Российской Федерации «О национальных целях развития Российской Федерации на период до 2030 года»;

Концепцией развития дополнительного образования до 2030 года (утверждена распоряжением Правительства РФ от 31.03.2022 № 678-р);

Стратегией развития воспитания в Российской Федерации на период до 2025 года (утверждена распоряжением Правительства Российской Федерации от 29.05.2015 № 996-р);

Планом мероприятий по реализации в 2021 - 2025 годах Стратегии развития воспитания в Российской Федерации на период до 2025 года (утвержден распоряжением Правительства Российской Федерации от 12.11.2020 № 2945-р);

Приказом Министерства образования и науки Российской Федерации от 23.08.2017 № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;

Приказом Министерства просвещения России от 09.11.2018 № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

Приказом Министерства просвещения РФ от 03.09.2019 № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;

Приказом Министерства Просвещения Российской Федерации от 30.09.2020 № 533 «О внесении изменений в порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденный приказом Министерства Просвещения Российской Федерации от 09.11.2018 № 196»;

Постановлением Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

Письмом Министерства образования и науки РФ от 18.11.2015 № 09-3242 «О направлении информации» (с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»).

Программа представляет несколько этапов, охватывающих отдельные базовые разделы программирования. Обучение начинается с изучения языка программирования C++, затем основных структур данных и алгоритмов работы с ними. Этапы предусматривают проведение промежуточной аттестации в виде самостоятельного решения задач по изученным разделам. Итоговая аттестация представляет собой самостоятельное решение задач по всем материалам, которые изучаются в рамках данной программы.

1.3 Адресат программы, требования к обучающимся, возрастные особенности группы

Программа «Основы программирования» адресована обучающимся 12-18 лет, не имеющих навыков программирования.

Набор на Программу осуществляется в соответствии с Порядком организации образовательной деятельности по дополнительным общеобразовательным общеразвивающим программам и Правилами приема и отчисления обучающихся автономной некоммерческой организации «Красноярский детский технопарк «Кванториум».

1.4 Педагогическая целесообразность

Выбор своего направления в IT-сфере - это очень сложная задача и поэтому программа построена таким образом, чтобы помочь учащимся заняться программированием, найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни, понять в каком направлении им хотелось бы развиваться дальше. Многие стремятся заняться информационными технологиями, не предполагая о всех возможностях этой области.

Изучение основ программирования связано с целым рядом умений и навыков, которые могут помочь в дальнейшей жизни не только будущим инженерам и программистам, но и тем, чья жизнь не будет напрямую связана с написанием программ.

Изучая программирование, обучающиеся получают глубокое понимание принципов работы компьютера, организации ввода, вывода и хранения информации, принципов построения диалоговых приложений, познают азы профессии программиста.

В рамках данной программы обучающиеся познакомятся с одним из самых сложных, но самым универсальным языком программирования C++, после чего увидят, как создаются различные программы и сами попробуют силы в разработке.

Содержание программы раскрывает процесс достижения результатов обучения по всем видам деятельности обучающихся: развитие интеллектуальных способностей и познавательных интересов через овладение технологиями создания программных продуктов.

Процесс обучения выстроен с учетом используемых современных образовательных технологий: информационные технологии обучения, технология проектного обучения, что отражено в дидактических принципах обучения (индивидуальность, доступность, преемственность, результативность и др.). Наряду с технологичностью обучения в образовательном процессе решаются задачи организации и управления. В соответствии с учебным планом программы на каждом этапе обучения разработаны занятия, наиболее эффективные для развития познавательного

потенциала обучающихся. Виды деятельности обучающихся отражены в предлагаемых формах организации обучения, что и определяет продолжительность по времени, динамичную структуру занятия, способы организации процесса обучения и его методическое оснащение. В техническом творчестве и проектной технологии ведущее место занимают интерактивные и активные методы обучения, обеспечивающие эффективное достижение поставленной цели обучения (в данной предметной области) и отслеживание результативности обучения на всех этапах.

1.5 Объем/срок освоения программы

Программа рассчитана на 1 год обучения. Годовая нагрузка на обучающегося составляет 144 часа.

1.6 Форма обучения, виды занятий и режим занятий

Учебные занятия проходят по очной форме обучения. Режим занятий – 2 раза в неделю по 2 академических часа (1 академический час 40 минут) с обязательным перерывом.

При проведении занятий используются комбинированные занятия – изложение нового материала, проверка пройденного материала, закрепление полученных знаний, самостоятельная работа.

При проведении занятий используются три формы работы:

- демонстрационная, когда обучающиеся слушают объяснения педагога и наблюдают за демонстрационным экраном или экранами компьютеров на ученических рабочих местах;
- фронтальная, когда обучающиеся синхронно работают под управлением педагога;
- самостоятельная, когда обучающиеся выполняют индивидуальные задания в течение части занятия.

Повторение и усвоение пройденного материала осуществляется через контрольные и проверочные работы, анализ полученных результатов.

Закрепление знаний, умений и навыков через постановку задачи и самостоятельную работу обучающегося под руководством педагога.

Применение полученных знаний и навыков через прикладную работу обучающегося, использующего на практике приобретенные компетенции.

В качестве основного метода обучения используется проектный метод.

1.7 Цель и задачи программы

Цель: формирование навыков алгоритмизации и программирования с использованием языка C++.

Задачи:

- сформировать навыки программирования на языке C++;
- сформировать знания об основных структурах данных;
- сформировать знания и навыки использования основных алгоритмов работы с данными;
- развить логическое и вычислительное мышление.

1.8 Планируемые результаты

Образовательные результаты

По результатам обучения, обучающиеся овладевают основами технологии программирования на языке C++, узнают основные структуры данных, а также изучат базовые алгоритмы.

Учащиеся будут знать:

- основы языка программирования C++;
- принципы процедурного программирования;
- принципы объектно-ориентированного программирования;
- основные алгоритмы и структуры данных.

Учащиеся будут уметь:

- создавать простые приложения с использованием языка программирования C++;
- работать со средой программирования Microsoft Visual Studio;
- осуществлять отладку разрабатываемых программ;
- применять изученные алгоритмы и структуры данных.

Сформированные навыки:

- создания алгоритмов;
- программирования на языке C++;
- применения различных структур данных и алгоритмов.
- умение правильно строить алгоритм и создавать программы разных типов.

Личностные результаты

Обучающиеся приобретут навыки самостоятельной организации своей деятельности; саморазвития и самовоспитания.

У обучающихся формируется готовность и способность к самостоятельной, творческой деятельности, к образованию, в том числе самообразованию; готовность к осознанному выбору будущей профессии.

Метапредметные результаты

Умение планировать; умение анализировать; умение алгоритмизировать.

1.9 Формы подведения итогов

По итогам каждого ключевого раздела проводится промежуточная аттестация в форме самостоятельной практической работы.

Аттестация по итогам освоения программы проводится в конце года и представляет собой практическую работу по пройденному материалу.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика
1	Знакомство с направлением обучения	4	4	0
1.1	Вводный урок. Правила и техника безопасности при работе с компьютером.	2	2	0
1.2	Что такое IT?	2	2	0
2	Язык программирования C++	8	4	4
2.1	Знакомство с C++.	2	2	0
2.2	Функции ввода/вывода. Переменные.	6	2	4
3	Типы данных	22	8	14
3.1	Целочисленные типы. Арифметические операции.	6	2	4
3.2	Типы с плавающей точкой. Арифметические операции.	6	2	4
3.3	Символьный тип данных.	6	2	4
3.4	Логический тип данных.	4	2	2
4	Конструкции языка	32	10	22
4.1	Условный оператор.	6	2	4
4.2	Логические операции.	6	2	4
4.3	Циклы. Цикл со счетчиком.	6	2	4
4.4	Циклы. Цикл с условием.	6	2	4
4.5	Оператор switch-case.	6	2	4
4.6	Промежуточная аттестация.	2	0	2
5	Массивы и функции	32	9	23
5.1	Одномерные массивы.	10	3	7
5.2	Многомерные массивы.	10	3	7
5.3	Функции.	12	3	9
6	Структуры	18	6	12
6.1	Работа с файлами	6	2	4
6.1	Структуры.	6	2	4
6.3	STL.	6	2	4
7	ООП	28	8	20
7.1	Классы.	8	2	6
7.2	Наследование.	6	2	4
7.3	Перегрузка.	6	2	4
7.4	Виртуальные методы.	6	2	4
7.5	Итоговая аттестация.	2	0	2
Итого		144	49	95

3. СОДЕРЖАНИЕ

1.1 Тема: Вводный урок. Правила и техника безопасности при работе с компьютером.

Теоретическая работа: Инструктаж по технике безопасности и правила поведения на занятиях.

1.2 Тема: Что такое IT.

Теоретическая работа: лекция о том, что включает в себя сфера информационных технологий, её роль в жизни общества, какие бывают профессии в этой сфере.

2.1 Тема: Знакомство с C++.

Теоретическая работа: Описание языка программирования C++, его возможности

Практическая работа: создание первой программы

2.2 Тема: Функции ввода/вывода.

Теоретическая работа: что такое функции ввода/вывода, потоковый ввод/вывод. Разбор примеров программ.

Практическая работа: применение изученных функций для решения задач.

3.1 Тема: Целочисленные типы. Арифметические операции.

Теоретическая работа: что такое целочисленные типы данных, какие арифметические операции существуют в языке C++. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

3.2 Тема: Типы с плавающей точкой. Арифметические операции.

Теоретическая работа: что такое и какие бывают численные типы данных с плавающей точкой. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

3.3 Тема: Символьный тип данных.

Теоретическая работа: что такое символьный тип данных, для чего используется. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

3.3 Тема: Логический тип данных.

Теоретическая работа: что такое логический тип данных, для чего используется. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

4.1 Тема: Условный оператор.

Теоретическая работа: что такое условный оператор, для чего применяется. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

4.2 Тема: Логические операции.

Теоретическая работа: логические функции И, ИЛИ, НЕ, для чего применяются. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

4.3 Тема: Циклы. Цикл со счетчиком.

Теоретическая работа: что такое цикл, для чего применяется, виды циклов, что такое цикл со счетчиком. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

4.4 Тема: Циклы. Цикл с условием.

Теоретическая работа: что такое цикл с условием, в каких ситуациях применяется. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

4.5 Тема: Оператор switch-case.

Теоретическая работа: что такое оператор switch-case, чем отличается от условия, в каких ситуациях применяется. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

4.6 Тема: Промежуточная аттестация.

Практическая работа: применение изученного материала для решения задач.

5.1 Тема: Одномерные массивы.

Теоретическая работа: что такое массив, элементы массива, индексы, для чего применяется. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

5.2 Тема: Многомерные массивы.

Теоретическая работа: что такое многомерный массив, для чего нужен. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

5.3 Тема: Функции.

Теоретическая работа: что такое и для чего нужны функции. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

6.1 Тема: Работа с файлами.

Теоретическая работа: для чего необходимо работать с файлами. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

6.2 Тема: Структуры.

Теоретическая работа: что такое структуры, как применяются структуры в программировании. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

6.3 Тема: STL.

Теоретическая работа: стандартная библиотека шаблонов, что включает в себя, как применять. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

7.1 Тема: Классы.

Теоретическая работа: что такое объектно-ориентированное программирование, что такое классы, чем отличаются от структур. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

7.2 Тема: Наследование.

Теоретическая работа: что такое наследование, классы-наследники, классы-родители. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

7.3 Тема: Перегрузка.

Теоретическая работа: перегрузка функций и операторов, что такое и для чего используется. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

7.4 Тема: Виртуальные методы.

Теоретическая работа: виртуальные функции и методы, что такое, для чего нужны, чем отличаются от перегрузок. Разбор примеров программ.

Практическая работа: применение изученного материала для решения задач.

7.5 Тема: Итоговая аттестация.

Практическая работа: применение изученного материала для решения задач.

4. СПИСОК ЛИТЕРАТУРЫ

Прата, С. Язык программирования C++: Лекции и упражнения / С. Прата. – М.: Вильямс, 2005. – 1097 с.

Шилдт, Г. Теория и практика C++ / Г.Шилдт. – СПб.:ВНУ, 1996. – 416 с.
cppreference.com [Электронный ресурс] – Режим доступа:
<https://en.cppreference.com/w/>

METANIT.COM – Сайт о программировании [Электронный ресурс] –
Режим доступа: <https://metanit.com/>

5. ИНФОРМАЦИОННОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОСНАЩЕНИЕ ДООП

Информационное оснащение:

Прата, С. Язык программирования С++: Лекции и упражнения / С. Прата. – М.: Вильямс, 2005. – 1097 с.

Артур О’Двайр Осваиваем С++17 STL / пер. с англ. А. Н. Киселева – М.: ДМК Пресс, 2019 – 352 с.: ил.

Бхаргава А. Грожаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. – СПб.: Питер, 2017. – 288 с.: ил. – (серия «Библиотека программиста»). (Электронная книга)

Материально-техническое оснащение:

Помещения для лабораторных занятий и самостоятельной работы обучающихся должны быть оснащены персональными компьютерами с установленной средой разработки программного обеспечения на языке объектно-ориентированного программирования Microsoft Visual Studio Community 2022 (и выше), Microsoft Office 2016 (и выше), с возможностью подключения к сети Интернет и доска с маркером или интерактивная доска. Помещение для занятий лекционного типа должно быть оснащено мультимедийным проектором и персональными компьютерами с установленной средой разработки программного обеспечения на языке объектно-ориентированного программирования Microsoft Visual Studio Community 2022 (и выше), Microsoft Office 2016 (и выше), с возможностью подключения к сети Интернет, доска с маркером или интерактивная доска.

6. Оценочные материалы

Критерии оценивания промежуточных проверочных работ и итоговой аттестации:

Критерий	Оценка
Программа выполняет требуемую функцию.	Зачет
Программа не выполняет требуемую функцию	Незачет

Программа считается освоенной, если в результате выполнения итоговой практической работы была получена оценка «Зачет».

7. ПРИЛОЖЕНИЯ

Приложение 1

Примеры задач для практического решения по темам:

2.1 Тема: Знакомство с C++.

Создать программу по шаблону, представленному в среде разработки.

2.2 Тема: Функции ввода/вывода.

Создать программу, выводящую в консоль сообщение «Привет, Мир!»

3.1 Тема: Целочисленные типы. Арифметические операции.

Создать программу, выводящую в консоль сумму, разность, произведение, целую часть от деления и остаток от деления двух целочисленных переменных, введенных пользователем.

3.2 Тема: Типы с плавающей точкой. Арифметические операции.

Создать программу, выводящую в консоль частное двух вещественных чисел, введенных пользователем.

3.3 Тема: Символьный тип данных.

Создать программу, выводящую в консоль символ, следующий в таблице ASCII, за введенным пользователем.

3.3 Тема: Логический тип данных.

Создать программу, выводящую в консоль результат выполнения трех базовых логических функций: И, ИЛИ, НЕ.

4.1 Тема: Условный оператор.

Создать программу, выводящую в консоль большее из двух целых чисел, введенных пользователем.

4.2 Тема: Логические операции.

Создать программу, выводящую в консоль наибольшее из трех целых чисел, введенных пользователем. При решении обязательно использовать функции И, ИЛИ, НЕ.

4.3 Тема: Циклы. Цикл со счетчиком.

Создать программу, выводящую столбиком в консоль все целые числа от 1 до введенного пользователем целого числа включительно.

4.4 Тема: Циклы. Цикл с условием.

Создать программу, реализующую повторный ввод данных, при вводе некорректных значений. Корректными являются натуральные числа.

4.5 Тема: Оператор switch-case.

Создать программу-калькулятор, выводящую в консоль результат выбранного пользователем математического действия (сумма, разность, произведение, частное), выполненного с двумя числами, введенными пользователем

5.1 Тема: Одномерные массивы.

Создать программу, в которой создается статический одномерный массив из 10 элементов, заполненный случайными двузначными числами, получившийся массив выводится в консоль.

5.2 Тема: Многомерные массивы.

Создать программу, в которой создается статический одномерный массив из 5 строк и 5 столбцов, заполненный случайными положительными однозначными числами. Главная диагональ массива заполняется нулями. Получившийся массив выводится в консоль.

5.3 Тема: Функции.

Реализовать функцию, выполняющую сортировку одномерного массива по убыванию.

6.1 Тема: Работа с файлами.

Создать программу, выводящую в консоль содержимое файла input.txt.

6.2 Тема: Структуры.

Создать структуру Figure, содержащую поля Name, Area, Perimeter. Создать объект структуры и заполнить его поля. Результат вывести в консоль.

6.3 Тема: STL.

Реализовать функцию, добавляющую в конец списка вещественных чисел элемент, значение которого равно среднему арифметическому всех его элементов.

7.1 Тема: Классы.

Создать класс Figure, содержащую поля Name, Area, Perimeter. Реализовать конструктор и деструктор класса, а также метод для вывода содержимого полей в консоль. Создать объект созданного класса и заполнить его поля. Результат вывести в консоль с помощью созданного метода.

7.2 Тема: Наследование.

Объявить класс «Фигура», содержащий поля, в которых хранится периметр и площадь фигуры. Создать 3 дочерних класса «Прямоугольник», «Треугольник» и «Окружность». В классе «Прямоугольник» должны быть поля, содержащие длину и ширину, а также в классе «Треугольник» должны быть поля, содержащие длины сторон. В классе «Окружность» должно быть поле, содержащее радиус окружности

7.3 Тема: Перегрузка.

Объявить класс «Фигура», содержащий поля, в которых хранится периметр и площадь фигуры. Создать 3 дочерних класса «Прямоугольник», «Треугольник» и «Окружность». В классе «Прямоугольник» должны быть поля, содержащие длину и ширину, а также В классе «Треугольник» должны быть поля, содержащие длины сторон. В классе «Окружность» должно быть поле, содержащее радиус окружности. Перегрузить оператор потокового вывода, для вывода в консоль информации об объектах этих классов.

7.4 Тема: Виртуальные методы.

Объявить класс «Фигура», содержащий поля, в которых хранится периметр и площадь фигуры. Создать 3 дочерних класса «Прямоугольник», «Треугольник» и «Окружность». В классе «Прямоугольник» должны быть поля, содержащие длину и ширину, а также В классе «Треугольник» должны быть поля, содержащие длины сторон. В классе «Окружность» должно быть поле, содержащее радиус окружности. Для всех дочерних классов должны быть реализованы методы, вычисляющие площадь и периметр.

Примеры заданий и тестовые данные для проведения промежуточного контроля:

Примеры заданий и тестовые данные для проведения промежуточного контроля успеваемости по разделу 1:

Задача:

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, оканчивающихся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 4. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число — сумму чисел, оканчивающихся на 4.

Пример тестовых данных:

Входные данные	Выходные данные
3 14 25 24	38

Задача:

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 6.

Пример тестовых данных:

Входные данные	Выходные данные
3 14 25 24	2

Задача:

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 6 и оканчивающихся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000. Программа должна вывести одно число: количество чисел, кратных 6 и оканчивающихся на 4.

Пример тестовых данных:

Входные данные	Выходные данные
3	2
14	
25	
24	

Примеры заданий для проведения итогового контроля:

Вариант 1

Создать класс «Трапеция», содержащий поля класса: координаты 4-х точек. Предусмотреть в классе конструктор, деструктор, метод проверки, является ли фигура равнобедренной трапецией, метод для вычисления и вывода сведений о фигуре: длин сторон, периметра и площади.

Для демонстрации работы с классом создать список из N (вводится пользователем) трапеций, заполнить его с клавиатуры, найти количество трапеций, у которых площадь больше средней площади.

Вариант 2

Создать класс «Книга», содержащий поля класса: название, автор, год издания. Предусмотреть в классе конструктор, деструктор, метод для вывода информации о книге в консоль.

Для демонстрации работы с классом создать вектор из N (вводится пользователем) книг, заполнить его с клавиатуры, отсортировать вектор в порядке возрастания года издания книг.