

ЗАДАНИЯ НА «УЧЕБНУЮ ПРАКТИКУ» НА 1-М КУРСЕ

ОБЩИЕ ТРЕБОВАНИЯ

Цели и задачи учебной практики по программированию на 1-м курсе состоят в приобретении и закреплении практических навыков процедурного и модульного программирования на языке Pascal (в среде программирования Turbo Pascal) и на языке C++ (в среде визуального программирования Microsoft Visual Studio .NET). При этом студент приобретает и закрепляет навыки создания программ, содержащих базовые управляющие структуры, процедуры, функции, а также использующих фундаментальные типы данных, массивы, указатели, ссылки, строки, структуры (записи), ввод-вывод данных, линейные связные списки и пр.

Общее задание на «Учебную практику» состоит из двух частей. В первой части студент согласно индивидуальному заданию разрабатывает набор программ на языке Pascal. Во второй части студент разрабатывает набор аналогичных программ на языке C++ (также в соответствии с индивидуальным заданием). Формулировки задач индивидуального задания берутся студентом из [1].

В процессе прохождения «Учебной практики» студент выполняет формализацию поставленных перед ним задач, изучает и/или разрабатывает алгоритмы решения этих задач и разрабатывает сами программы, решающие поставленные задачи.

«Учебная практика» обычно проводится в летний период, как правило, в течение трех первых недель сразу после летней сессии (допускается проведение «Учебной практики» в течение учебного года).

По завершении «Учебной практики» студент предоставляет отчет в формате *Microsoft Word*, содержащий *формулировки задач индивидуального задания* и *тексты программ*, реализующих решение этих задач, а также сдает дифференцированный зачет по результатам выполнения индивидуального задания.

Студент, завершивший прохождение «Учебной практики», должен быть подготовлен к самостоятельной разработке программ на языках программирования Pascal и C++, использующих парадигмы структурного, процедурного и модульного программирования.

ПЕРЕЧЕНЬ ОБЩИХ ФОРМУЛИРОВОК ЗАДАЧ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

Часть I. Язык программирования PASCAL (34 ч.)

Алгоритмы простой структуры (10 ч.)

1. Вычисление в цикле таблицы значений функции с одним аргументом. Составление *алгоритма* и *программы* вычисления и вывода таблицы значений функции $f(x)$ для n значений аргумента x , равномерно распределенных на отрезке $[a; b]$ (задача 1.2).
2. Нахождение суммарных значений в цикле. Составление *алгоритма* и *программы* для вычисления в цикле суммарного результата и вывода его на экран (задача 1.3.3).
3. Нахождение экстремальных элементов в одномерном массиве. Составление *алгоритма* и *программы* нахождения экстремального элемента и/или его порядкового номера в одномерном массиве (или массивах) из n элементов и вывода результата поиска на экран (задача 1.4.3).

4. Вычисление корня функционального уравнения. Разработка *алгоритма* и *программы* для решения уравнения $f(x) = 0$ на отрезке $[a; b]$ с заданной точностью методом деления отрезка пополам и вывода решения на экран (задача 1.7).
5. Логическая проверка попадания точки в заданную область. Разработка *программы* проверки попадания точки в заданную область и вывода результатов проверки на экран (задача 1.9.4).

Алгоритмы более сложной структуры (6 ч.)

6. Обработка массивов произвольной длины. Разработка *программы* для получения и вывода на экран указанных в условии результатов (задача 2.5.2).
7. Табулирование функции двух аргументов. Разработка *программы* вычисления и вывода на экран таблицы значений функции двух аргументов с использованием матрицы для хранения значений функции (задача 3.2).

Процедуры и функции (6 ч.)

8. Использование массивов в качестве аргументов функции. Составление *программы*, использующей функцию, аргументами которой являются массивы (одномерные или двумерные) (задача 4.2.3).
9. Разработка многомодульной программы с выделением подпрограмм. Составление *программы* на основе предварительно выделенных процедур и функций (задача 4.3.3).

Строки, записи, указатели и файловый ввод-вывод (12 ч.)

10. Ввод и обработка массива записей. Составление *программы*, выполняющей ввод с клавиатуры массива записей о студентах некоторого факультета и обработку введенного массива записей согласно индивидуальному заданию (задача 5.2.1).
11. Создание, вывод на экран и уничтожение линейного связного списка записей. Составление *программы*, выполняющей создание линейного связного списка записей, вывод созданного списка на экран и уничтожение его из памяти перед завершением программы. При этом использовать указатели и динамическое распределение памяти (исходные данные о структуре записей списка см. в задаче 5.2.2).

Часть II. Язык программирования C++ (34 ч.)

Алгоритмы простой структуры (10 ч.)

1. Вычисление в цикле таблицы значений функции с одним аргументом. Составление *программы* вычисления и вывода на экран таблицы значений функции $f(x)$ для n значений аргумента x , равномерно распределенных на отрезке $[a; b]$ (задача 1.2).
2. Обработка одномерных массивов с единственным циклом. Составление *программы* с единственным циклом для обработки одномерных массивов и вывода результатов обработки на экран (задача 1.4.1).
3. Вычисление таблицы значений функции одного аргумента с выбором формулы. Разработка *программы* вычисления n значений функции $y = f(x)$ для x , изменяющегося от x_1 с шагом Δx и вывода результатов вычисления на экран в виде таблицы с использованием псевдографики (задача 1.6).
4. Сложное условие завершения цикла. Составление *программы*, вычисляющей значения функции и выводящей на экран результаты вычисления до тех пор, пока не будет пройдена некоторая характерная точка графика функции. Аргумент задается в виде возрастающей арифметической прогрессии (задача 1.8).
5. Логическая проверка попадания точки в заданную область. Разработка *программы* проверки попадания точки в заданную область и вывода результатов проверки на экран (задача 1.9.4).

Алгоритмы более сложной структуры (6 ч.)

6. Обработка массивов произвольной длины. Разработка *программы* для получения и вывода на экран указанных в условии результатов (задача 2.5.2).
7. Сложное условие завершения цикла при работе с матрицами. Составление *программы*, предусматривающей завершение обработки матрицы при выполнении указанного в задаче условия и выполняющей вывод результатов обработки на экран (задача 2.6).

Функции (6 ч.)

Замечание. Здесь и далее консольный ввод-вывод в программах следует выполнять на русском языке, используя преобразование кодировок. При этом желательно применять стандартный потоковый ввод-вывод в стиле языка C++.

8. Вычисление заданной функции разложением ее в ряд с использованием рекурсии. Составление *программы* вычисления заданной математической функции путем разложения ее в ряд. Программа должна применять рекурсивную функцию для организации вычислений (задача 4.2.1).
9. Использование массивов в качестве аргументов функции. Составление *программы*, использующей функцию, аргументами которой являются массивы (одномерные или двумерные) (задача 4.2.3).
10. Разработка многомодульной программы с выделением функций. Разработка *программы* на основе предварительно выделенных функций (задача 4.3.3).

Строки, структуры, связанные списки и файловый ввод-вывод (12 ч.)

11. Обработка линейного списка структур с применением нестандартных средств консольного ввода-вывода. Разработка *программы* выполняющей над связным списком структур следующие операции: ввод списка с клавиатуры; вывод списка на экран; добавление новых структур в начало списка, в его конец или в заданную позицию списка; удаление из списка структуры по ее номеру; сохранение в файле и чтение из файла всего списка структур; определение длины списка; обработка списка согласно индивидуальному заданию. Программа обязательно должна обеспечивать возможность выбора пользователем нужной ему операции через *текстовое меню*. Каждая операция должна быть реализована в программе в виде отдельной функции (задача 5.2.2).

ФОРМА ОТЧЕТНОСТИ ПО ЗАВЕРШЕНИИ УЧЕБНОЙ ПРАКТИКИ

По завершении «Учебной практики» студент должен при себе иметь:

1. Завершенные проекты программ на языках Pascal и C++ по выполнению задач из индивидуального задания с исходными (и заголовочными при использовании языка C++) файлами единиц трансляции проекта (на дискете).
2. Отчет по учебной практике в электронном виде (на дискете).
3. Отчет по учебной практике в печатном виде (твердая копия отчета).

СТРУКТУРА ОТЧЕТА

1. Титульный лист.
2. Содержание отчета.
3. Введение, отражающее цели и задачи учебной практики.
4. Номер варианта индивидуального задания.
5. Часть I – содержит формулировки задач индивидуального задания и тексты исходных файлов программ на языке Pascal.
6. Часть II – содержит формулировки задач индивидуального задания и тексты ис-

ходных и заголовочных файлов программ на языке C++.

7. Список литературы, используемой при выполнении индивидуального задания.

ОСНОВНЫЕ ТРЕБОВАНИЯ К ПРОГРАММАМ

Программы должны удовлетворять следующим основным требованиям:

1. Программы должны нормально (не аварийно) выполняться и завершаться в операционной системе Windows 2000.
2. Программы должны иметь удобный и интуитивно понятный интерфейс.

ЛИТЕРАТУРА

ОСНОВНАЯ ЛИТЕРАТУРА

1. **Сборник** задач по базовой компьютерной подготовке. / В. С. Зубов, И. Н. Котарова, О. Г. Батасова и др.; Составитель И. Н. Котарова. – М.: Изд-во МЭИ, 1998.
2. **Немнюгин С. А.** Turbo Pascal / С. А. Немнюгин. – СПб.: Питер, 2002.
3. **Климова Л. М.** Pascal 7.0. Практическое программирование. Решение типовых задач: Учеб. пособие для вузов / Л. М. Климова. – М.: УДИЦ-ОБРАЗ, 2000.
4. **Липпман С.**, Лажоие Ж. Язык программирования С++. Вводный курс, 3-е изд. / Пер. с англ. – СПб. – М.: Невский Диалект – ДМК Пресс, 2003. – 1104 с.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

5. **Марченко А. И.** Программирование в среде Turbo Pascal 7.0.: Базовый курс. / А. И. Марченко, Л. А. Марченко. – 5-е изд. – Киев: Век+, 1999.
6. **Б. Страуструп.** Язык программирования С++, спец. изд. / Пер. с англ. – М.: СПб.: «Издательство БИНОМ» – «Невский Диалект», 2002. – 1099 с.
7. **Т. Кормен**, Ч. Лейзерсон, Р. Ривест. Алгоритмы: построение и анализ / Пер. с англ. под ред. А. Шеня. – М.: МЦНМО, 2002. – 960 с.
8. **С. Прата.** Язык программирования С++: Лекции и упражнения. / Пер. с англ. – СПб.: ООО «ДиаСофтЮП», 2003. – 1104 с.