

DOI: 10.24888/2500-1957-2024-2-19-25

УДК
371.322.2**РАЗРАБОТКА ВАРИАТИВНЫХ ТИПОВЫХ ЗАДАНИЙ ПО
ПРОЦЕДУРНОМУ ПРОГРАММИРОВАНИЮ В СРЕДНЕЙ
ШКОЛЕ**

Лобов Денис Владимирович к.ф.-м.н, доцент ldenis@petrsu.ru г. Петрозаводск	Петрозаводский государственный университет
Васькин Виталий Александрович студент vitalikvaskin11@yandex.ru г. Петрозаводск	Петрозаводский государственный университет
Логинов Дмитрий Владимирович к.ф.-м.н, доцент logindm@mail.ru г. Петрозаводск	Петрозаводский государственный университет

Аннотация. Авторами работы приведён подход, который упрощает освоение темы «Процедурное программирование» учащимися школ. Подход включает в себя разработку вариативных типовых заданий по процедурному программированию с учётом указанных в статье требований, а также дифференцированного обучения и системно-деятельностного подхода. Разработаны шесть основных типов заданий для обучения: составление подпрограмм, их анализ и поиск ошибок, в том числе в готовых алгоритмах. В статье приводятся примеры составления типовых заданий. Данный подход способствует развитию умения разбивать сложную проблему на более мелкие и легко выполнимые задачи.

Ключевые слова: информатика, преподавание информатики, программирование, процедурное программирование, типовые задачи процедурного программирования.

В Федеральном государственном образовательном стандарте (ФГОС) среднего общего образования (СОО) в требованиях к предметным результатам освоения базового и углубленного курса «Информатика» присутствуют следующие требования: владение универсальным языком программирования высокого уровня (по выбору), представление о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции (Федеральный государственный образовательный стандарт среднего общего образования, 2012). После изучения углубленного курса «Информатика» обучающиеся должны уметь работать с библиотеками программ, что подразумевает необходимость изучения процедурного программирования.

В федеральной рабочей программе среднего общего образования по информатике как базового, так и углубленного уровня тема, являющаяся продолжением структурной части императивной парадигмы программирования – процедурное программирование, входит в обязательное изучение (Федеральная рабочая программа учебного предмета «Информатика. Базовый уровень»). В частности, в 9-м классе рассматриваются подпрограммы для исполнителей среды «КуМир» (Робот, Черепашка и т. д.). При этом в 11-м классе тема «Подпрограммы» рассматривается на конкретном языке программирования.

МЕТОДИЧЕСКИЕ АСПЕКТЫ ОБУЧЕНИЯ МАТЕМАТИКЕ И ИНФОРМАТИКЕ В СИСТЕМЕ ОБЩЕГО ОБРАЗОВАНИЯ

Кроме того, некоторые задания единого государственного экзамена (ЕГЭ) по предмету «Информатика» предполагают использование подпрограмм (процедур и функций). В разделе «Перечень элементов содержания» кодификатора ЕГЭ в компонентах «Основные конструкции языка программирования (Кодификатор). Система программирования и «Основные этапы разработки программ. Разбиение задачи на подзадачи» прописано, что обучающийся должен уметь разрабатывать алгоритмы и программы с использованием подпрограмм».

С учётом указанных требований, а также дифференцированного обучения и системно-деятельностного подхода к обучению стоит доработать и расширить разработанный ранее набор типовых индивидуальных заданий, включив раздел «процедурное программирование» (Васькин, 2022).

Стоит отметить, что приведённый в (Васькин, 2022) обобщённый список заданий для проверки знаний охватывал два раздела содержательной линии «Алгоритмизация и программирование»:

- основы алгоритмизации, способы описания алгоритмических конструкций (например, прямая и обратная трассировки готового алгоритма, представленного в различных формах; составление блок-схем и словесно-формульного описания алгоритмов, содержащих линейные, ветвящиеся, повторяющиеся конструкции и все их возможные комбинации и другие);

- начала программирования на языке Pascal (исправление ошибок в коде (синтаксические и семантические ошибки), прямая и обратная трассировка программы и другие).

На основе анализа наиболее распространённых учебно-методических комплектов (УМК) за 10-й и 11-й классы (Поляков, 2013), (Босова, 2017), (Семакин, 2015) был расширен обобщённый список заданий по алгоритмизации и программированию (Васькин, 2022) с учётом добавления процедурного программирования.

Основные типы заданий для обучения по теме: «Процедурное программирование»:

1. Прямая и обратная трассировка подпрограммы.
Трассировка алгоритма — это последовательный проход по всем шагам алгоритма при заданных входных или выходных данных. Данный тип задач позволяет научить обучающихся определить цель алгоритма; правильность его работы; принадлежность входных данных разрешённому классу; входные параметры на основе выходных.
2. Определение цели данной подпрограммы.
Анализ и синтез алгоритма для понимания преобразования входных данных в выходные.
3. Исправление ошибок в коде подпрограмм (синтаксические и семантические ошибки).
Поиск синтаксических ошибок базируется на уровне знаний грамматики языка программирования. Поиск семантических ошибок не зависит от конкретного языка и направлен на исправление алгоритма таким образом, чтобы результат выполнения соответствовал цели решаемой задачи.
4. Выбор минимально необходимых параметров подпрограммы.
Анализ алгоритма подпрограммы для определения минимально необходимого количества входных и выходных параметров.
5. Составление подпрограмм на основе заданных: алгоритма, подпрограммы, программы.
Анализ алгоритма и вычленение рутин ввода, преобразования и вывода данных для создания подпрограмм.
6. Преобразование процедуры в функцию и наоборот.

В большинстве языков программирования присутствуют два типа подпрограмм: процедура и функция. Определение цели, входных и выходных параметров для возможности преобразования из одного типа подпрограмм в другой.

Среди стандартных задач как базового, так и углубленного уровней присутствует задача поиска простых чисел. Рассмотрим весь обобщённый список заданий на примере данной задачи.

Идея метода: простым числом является такое натуральное число, которое делится только на единицу и самого себя. Для решения данной задачи проверим условие отсутствия таких чисел, меньших N , на которые N делится без остатка.

Ниже приведён алгоритм и программа на языке Pascal. И в алгоритме, и в программе используются процедура и функция.

Алгоритм 1	Программа 1
<pre> 0. алгоритм ПростЧисло(параметры NL, flagL) 1. Начало 2. Если NL = 1, то flagL = 0 3. Цикл i=2,NL-1 3.1. flagL = 1 3.2. если NL mod i = 0, то 3.2.1. flag = 0 3.2.2. выход из цикла 4. Конец 0. Начало 1. Ввод N 2. ПростЧисло(N, flag) 3. Если flag = 1, то вывод «N - простое!» 4. иначе вывод «Число N - не простое!» 5. Конец </pre>	<pre> var N: longint; flag: byte; Procedure PrimeNumP(NL:longint;var flagL: byte); var i: longint; begin if(NL=1) then flagL:=0; for i:=2 to NL-1 do begin flagL:=1; if(NL mod i = 0) then begin flagL:=0; break; end; end; end; begin readln(N); PrimeNumP(N, flag); if(flag = 1) then writeln('N-прост!') else writeln('N - не простое!'); readkey; end. </pre>
Алгоритм 2	Программа 2
<pre> 0. алгоритм ПростЧисло(параметр NL) 1. Начало 2. Если NL = 1, то flagL = 0 3. Цикл i=2,NL-1 3.1. flagL = 1 3.2. если NL mod i = 0, то 3.2.1. flagL = 0 3.2.2. выход из цикла 4. ПростЧисло = flagL 5. Конец 0. Начало 1. Ввод N 2. Если ПростЧисло(N) = 1, то 2.1 вывод «N - простое!» иначе 2.1 вывод «Число N - не простое!» 3. Конец </pre>	<pre> var N: longint; flag: byte; Function PrimeNumF(NL: longint): byte; var i: longint; flagL: byte; begin if(NL=1) then flagL:=0; for i:=2 to NL-1 do begin flagL:=1; if(NL mod i = 0) then begin flagL:=0; break; end; end; PrimeNumF:= flagL; end; begin readln(N); if(PrimeNumF(N)=1) then writeln('N-прост!') else writeln('N - не простое!'); readkey; end. </pre>

На основе выше приведённого решения был сформулирован список конкретных заданий.

Прямая и обратная трассировка подпрограммы

1. Дан алгоритм 1. Определить результат работы алгоритма при $N = 3$.
2. Используя трассировочную таблицу для процедуры PrimeNumP программы 1, нужно определить, сколько итераций потребуется для выхода из цикла при $N = 3$.

**МЕТОДИЧЕСКИЕ АСПЕКТЫ ОБУЧЕНИЯ МАТЕМАТИКЕ И ИНФОРМАТИКЕ
В СИСТЕМЕ ОБЩЕГО ОБРАЗОВАНИЯ**

3. Используя трассировочную таблицу (табл. 1) для процедуры PrimeNumP программы 1, необходимо определить вводимое число.
4. Используя трассировочную таблицу (таблица 1) для процедуры PrimeNumP программы 1, нужно подсчитать сколько раз условие $\text{if}(N \bmod i = 0)$ будет истинным при $N = 3$.

Таблица 1. Трассировка процедуры PrimeNumP.

Шаг	Операция	NL	i	flagL	Условие
2	if (NL = 1) then	3	-	-	3=1, нет
3	for i := 2 to NL - 1 do	3	2	-	
4	flagL := 1	3	2	1	
5	if (NL mod i = 0) then	3	2	1	3 mod 2=0, нет
6	flagL:= 0	3	2	1	
7	break	3	2	1	

Определение цели данной подпрограммы

1. Дана программа 1 (или программа 2). Вычислительные операции выполняются в процедуре PrimeNumP (функции PrimeNumF). Необходимо определить цель данной процедуры (функции) и ответить на вопрос: какие переменные идут на вход, а какие на выход?
2. Дан алгоритм 1 (алгоритм 2). Известно, что процедура ПростЧисло (или функция ПростЧисло) проверяет простое ли число, подаваемое как входной параметр. Нужно определить цель данного алгоритма.

Исправление ошибок в коде подпрограмм (синтаксические и семантические ошибки)

<p>1. Ниже приведена программа на языке Pascal, проверяющая простое ли число. При написании кода допущены синтаксические ошибки. Нужно найти и исправить их.</p>	<p>2. Ниже приведен алгоритм, проверяющий простое ли число. При его составлении допущены семантические (логические) ошибки. Необходимо найти и исправить их.</p>
<pre>var N: longint; flag: byte; Function PrimeNumF(NL: longint); var iL: longint; flagL: byte; begin if(NL==1) then flagL:=0; for i:=2 to NL-1 do begin flagL=1; if(NLmodi = 0) then begin flagL:=0; break; end; end; PrimeNumF = flagL; end</pre>	<p>0. алгоритм ПростЧисло(параметры NL, flagL)</p> <ol style="list-style-type: none"> 1. Начало 2. Если NL = 1, то flagL = 0 3. Цикл i=2,NL-1 <ol style="list-style-type: none"> 3.1. flagL = 1 3.2. если NL div i = 0, то <ol style="list-style-type: none"> 3.2.1. flag = 1 3.2.2. выход из цикла 4. Конец <p>0. Начало</p> <ol style="list-style-type: none"> 1. Ввод N 2. ПростЧисло(N, flag) 3. Если flag = 1, то вывод «N – простое!»

<pre>begin readln(N); if(PrimeNumF)=1) then writeln('N-прост!') else writeln('N - не простое!'); readkey; end.</pre>	<p>4. иначе вывод «Число N – не простое!» 5. Конец</p>
--	--

Выбор оптимально необходимых параметров подпрограммы

Составление подпрограмм на основе заданных условий или алгоритма.

<p>1. Дан код программы, написанный на языке Pascal. Необходимо проанализировав процедуру PrimeNumP и её вызов, написать для неё формальные и локальные параметры.</p>	<p>1. Используя приведённый ниже алгоритм, необходимо написать программу на языке программирования Pascal. Примечание: основные вычисления программы необходимо реализовать в подпрограмме, в которой на вход подаётся два параметра [предполагается использование процедуры] (один параметр [предполагается использование функции]).</p>
<pre>{Для усложнения задания можно скрыть строчку с описанием глобальных параметров самой программы} var N: longint; flag: byte; Procedure PrimeNumP(?); var ?; begin if(NL=1) then flagL:=0; for i:=2 to NL-1 do begin flagL:=1; if(NL mod i = 0) then begin flagL:=0; break; end; end; end; begin readln(N); if(PrimeNumF(?)=1) then writeln('N-прост!') else writeln('N - не простое!'); readkey; end.</pre>	

Создание новой подпрограммы на основе имеющегося алгоритма/подпрограммы/программы

Дан алгоритм 1 (алгоритм 2). Написать его реализацию на языке Pascal.

Преобразование процедуры в функцию и наоборот

Дана программа 1 (программа 2). Основные вычисления производятся в процедуре PrimeNumP (функции PrimeNumF). Нужно переписать программу, преобразовав процедуру в функцию (в процедуру).

Разбиение программы на подпрограммы ввода, вывода и преобразования данных

МЕТОДИЧЕСКИЕ АСПЕКТЫ ОБУЧЕНИЯ МАТЕМАТИКЕ И ИНФОРМАТИКЕ В СИСТЕМЕ ОБЩЕГО ОБРАЗОВАНИЯ

Дан код программы 1 (программы 2), написанный на языке Pascal. Необходимо разбить программу на подпрограммы, реализующие ввод, вывод и преобразование данных.

Заключение

Был проведён анализ наиболее распространённых учебно-методических комплектов (УМК) для основной и средней школ (Поляков, Босова, Семакин). На основе данного анализа был расширен ранее разработанный обобщённый список заданий по алгоритмизации и программированию, в который была включена тема «Процедурное программирование».

Одним из преимуществ изучения темы «Процедурное программирование» является развитие у школьников умения разбивать сложную проблему на более мелкие и легко выполнимые задачи. Понимая пошаговый характер процедурного программирования, школьники развивают навыки решения проблем и учатся подходить к системному решению задач не только в программировании, но и в любых других профессиональных сферах деятельности.

Список литературы

- Босова Л.Л. Информатика. 11 класс. Базовый уровень / Л.Л. Босова, А.Ю. Босова. М.: БИНОМ. Лаборатория знаний, 2017.
- Васькин В.А. Разработка авторских типовых заданий для проверки знаний по алгоритмизации и программированию в 8-м классе школы // StudArctic Forum. Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Петрозаводский государственный университет». 2022. Т. 7. №. 2. С. 73-80.
- Кодификатор проверяемых требований к результатам освоения основной образовательной программы среднего общего образования и элементов содержания для проведения единого государственного экзамена по информатике [Электронный ресурс]. URL: <https://fipi.ru/ege/demoversii-specifikacii-kodifikatory> (Дата обращения: 28.11.2023)
- Поляков К.Ю. Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 2 / К.Ю. Поляков, Е.А. Еремин. М.: БИНОМ. Лаборатория знаний. 2013.
- Семакин И.Г. Информатика. Базовый уровень: учебник для 10 класса / И.Г. Семакин, Е.К. Хеннер, Т.Ю. Шеина. 4-е изд. М.: БИНОМ. Лаборатория знаний, 2015.
- Федеральный Государственный образовательный стандарт среднего общего образования: утверждён приказом Министерства образования и науки Российской Федерации 17 мая 2012 г. № 413 / Министерство образования и науки Российской Федерации. Москва: 2012г.
- Федеральная рабочая программа учебного предмета «Информатика. Базовый уровень» [Электронный ресурс]. URL: <https://edsoo.ru/rabochie-programmy/> (дата обращения: 28.11.2023)

DEVELOPMENT OF VARIABLE STANDARD TASKS IN PROCEDURAL PROGRAMMING IN HIGH SCHOOL

Lobov D. V. Ph.D. (Physics and math), assistant professor ldenis@petsu.ru Petrozavodsk	Petrozavodsk State University
Vaskin V. A. student vitalikvaskin11@yandex.ru	Petrozavodsk State University

Petrozavodsk	
Loginov D. V.	Petrozavodsk State University
Ph.D. (Physics and math), assistant professor logindm@mail.ru Petrozavodsk	

Abstract. The authors of the work present an approach that simplifies the development of the topic “Procedural Programming” by school students. The approach includes the development of variable standard tasks for procedural programming, taking into account the requirements specified in the article, as well as differentiated training and a system-activity approach. Six main types of tasks for training have been developed: compiling subroutines, analyzing them and searching for errors, including in ready-made algorithms. The article provides examples of composing typical tasks. This approach helps develop the ability to break a complex problem into smaller, easily manageable tasks.

Keywords: computer science, computer science teaching, programming, procedural programming, typical problems of procedural programming

References

- Bosova, L. L., Bosova, A. Yu. (2017). Informatika. 11 klass. Bazovyj uroven'. Moscow: BINOM. Laboratoriya znaniy. (In Russ.)
- Federal'nyj Gosudarstvennyj obrazovatel'nyj standart srednego obshchego obrazovaniya: utverzhdyon prikazom Ministerstva obrazovaniya i nauki Rossijskoj Federacii 17 maya 2012 g. № 413 / Ministerstvo obrazovaniya i nauki Rossijskoj Federacii. (In Russ.)
- Federal'naya rabochaya programma uchebnogo predmeta «Informatika. Bazovyj uroven'». <https://edsoo.ru/rabochie-programmy> (In Russ.)
- Kodifikator proveryaemyh trebovanij k rezul'tatam osvoeniya osnovnoj obrazovatel'noj programmy srednego obshchego obrazovaniya i elementov soderzhaniya dlya provedeniya edinogo gosudarstvennogo ekzamina po informatike. <https://fipi.ru/ege/demoversii-specifikacii-kodifikator>. (In Russ.)
- Polyakov, K. Yu., Yeremin, E. A. (2013). Informatika. Uglublennyj uroven': uchebnik dlya 10 klassa: v 2 ch. Ch. 2. BINOM. Laboratoriya znaniy. (In Russ.)
- Semakin, I. G., Henner, E. K., Sheena, T. Y. (2015). Informatika. Bazovyj uroven': uchebnik dlya 10 klassa. Moscow: BINOM. Laboratoriya znaniy. (In Russ.)
- Vaskin, V. A. (2022). Razrabotka avtorskih tipovyh zadaniy dlya proverki znaniy po algoritmizacii i programmirovaniyu v 8-m klasse shkoly. StudArctic Forum. Federal'noe gosudarstvennoe byudzhethnoe obrazovatel'noe uchrezhdenie vysshego professional'nogo obrazovaniya «Petrozavodskij gosudarstvennyj universitet», 7(2), 73-80. (In Russ.)

Статья поступила в редакцию 28.03.2024
Принята к публикации 10.06.2024