

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таскаев Сергей Валерьевич
Должность: Ректор
Дата подписания: 08.09.2025 09:30:14
Уникальный программный ключ: Южно-Уральский государственный университет
0919341810985335075486193098887821373

Министерство науки и высшего образования Российской Федерации
Южно-Уральский государственный университет
Институт естественных и точных наук
Факультет математики, механики и компьютерных технологий
Кафедра прикладной математики и программирования

004.4(07)
Д304

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Методические указания по выполнению курсовых работ

Челябинск
Издательский центр ЮУрГУ
2020

УДК 004.434(075.8)
Д304

Одобрено
учебно-методической комиссией института естественных и точных наук

Рецензент
Е.М. Сартасов

Д304 **Языки программирования:** методические указания по выполнению курсовых работ / сост. А.К. Демидов. – Челябинск: Издательский центр ЮУрГУ, 2020. – 38 с.

В методическом издании рассматриваются этапы разработки курсовой работы по дисциплине «Языки программирования», приводятся критерии оценки, требования к оформлению и содержанию пояснительной записки и презентации.

Методическое издание содержит перечень тем для курсовых работ, рекомендации по разработке алгоритма и оформлению схем алгоритмов, по реализации, тестированию и отладке программы, порядок защиты курсовой работы, структура доклада и презентации. В приложениях приводятся примеры оформления основных частей пояснительной записки, а также пример презентации.

Методическое издание предназначено для студентов направлений 01.03.02 «Прикладная математика и информатика».

УДК 004.434(075.8)

© Издательский центр ЮУрГУ, 2020

1 ЭТАПЫ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Курсовая работа по дисциплине «Языки программирования» представляет собой учебно-практическое исследование, предназначенное для систематизации, углубления и закрепления знаний, полученных студентом в процессе изучения конкретной дисциплины в соответствии с учебным планом.

Перечень этапов, распределение времени и баллов показаны в таблице 1.

Таблица 1 — Продолжительность и трудоемкость этапов

Наименование этапа	Продолжительность	Часы	Баллы	Результат работы
постановка задачи	2 недели	2	5	Раздел 1 Постановка задачи
разработка алгоритма	4 недели	10	20	Раздел 2 Разработка алгоритма
реализация программы	3 недели	12	15	Исходный код
тестирование программы, улучшение и исправление ошибок	3 недели	5	20	Исполняемый код
оформление программной документации и отчета по курсовой работе	2 недели	5	10	Раздел 3 Руководство пользователя Пояснительная записка
защита курсовой работы	2 недели	3	30	Презентация к докладу
ВСЕГО	16 недель	41	100	

2 ТЕМЫ ДЛЯ КУРСОВЫХ РАБОТ

В таблице 2 приведен краткий перечень тем курсовых работ, более подробная информация на сайте [1]. Используемые обозначения в колонке «Графика/текст»: Г — графика с использованием библиотеки winBGIm, Т — текстовый режим с использованием библиотеки conio, Т/Г — при разработке программы можно выбрать текстовый или графический режим. Обозначения в колонке «Вид игры»: О — игра для одного человека (головоломка, пасьянс), 2Ч, 2+Ч — игра для двух или более человек на одном компьютере, 2ЧС, 2+ЧС — игра для двух или более человек по сети, каждый человек за своим компьютером, ЧК — игра для человека с компьютером (описание стратегии компьютера по ссылкам, при отсутствии ссылок - случайный ход по правилам игры). В колонке сложность указана предположительная сложность программы: 0 — объем кода от 50 до 100 строк (максимальная оценка «хорошо» при выполнении всех критериев получения этой оценки), 1 — от 100 до 200 строк, 2 — от 200 до 400 строк, 3 — от 200 строк, поиск и применение специальных алгоритмов (возможно сокращение требований из первоначальной постановки задачи по согласованию с руководителем без влияния на итоговую оценку).

Таблица 2 — Темы курсовых работ

№	Темы курсовых работ	Ссылки	Текст/ гра- фика	Вид игры	Слож- ность
1	Векторный графический редактор		Г	—	3
2	Графический редактор для детей		Г	—	2
3	Ежедневник (список дел)		Г	—	1
4	Игра «Lines»		Г	О	1
5	Игра «Арканоид»		Г	О	1
6	Игра «Бильярд»		Г	О	1
7	Игра «Волейбол»		Г	2Ч	2
8	Игра «Домино»		Г	ЧК	2
9	Игра «Дурак»		Г	ЧК	2
10	Игра «Змейка»		Г	О	1
11	Игра «Как стать миллионером»		Г	О	1
12	Игра «Кладовщик» (сокобан)		Г	О	1
13	Игра «Крестики-нолики» на поле 3x3		Г	2Ч	0
14	Игра «Восьмерки»	[2]	Г	ЧК	2
15	Игра «Монополия»		Г	2+Ч	2
16	Пасьянс «Косынка»		Г	О	1
17	Пасьянс «Паук»		Г	О	1
18	Игра «Пифагор» (танграм)		Г	О	1
19	Игра «Видеопокер»	[3]	Г	О	1
20	Игра «Покер на костях»	[4]	Г	2+Ч	1
21	Игра «Рэндзю»		Г	2Ч	1
22	Игра «Сапер»		Г	О	0
23	Игра «Тетрис» (классический)		Г	О	2
24	Игра «Columns»		Г	О	2
25	Игра «Математический тетрис» (числа, заданная сумма в соседних клетках)		Г	О	1
26	Игра «Червы»		Г	ЧК	2
27	Игра «Шахматы»		Г	2Ч	3
28	Игра «Шашки»		Г	2Ч	2
29	Календарь		Т/Г	—	1
30	Инженерный калькулятор (мат.функции, скобки)		Г	—	2

Продолжение таблицы 2

№	Темы курсовых работ	Ссылки	Текст/ гра- фика	Вид игры	Слож- ность
31	Простой калькулятор (вещ. числа, +, -, *, /, %, =, MS, MR)		Г	—	2
32	Клавиатурный тренажер		Т/Г	О	0
33	Настольная игра (змеи и лестницы, эстафета)		Г	2+Ч	1
34	Проводник (показ списка файлов и папок и переход в другие папки)		Г	—	1
35	Классный журнал (оценки, посещаемость)		Т/Г	—	1
36	Редактор html-страниц (подсветка и закрытие тегов)		Т/Г	—	3
37	Редактор схем алгоритмов		Г	—	3
38	Редактор генеалогического дерева		Г	—	3
39	Редактор иконок в формате .ico для сайтов		Г	—	2
40	Редактор карт (география)		Г	—	3
41	Редактор радио-схем		Г	—	3
42	Редактор символов (растровых шрифтов)		Г	—	2
43	Редактор таблиц (заполнение ячеек)		Т/Г	—	2
44	Рисование графиков математических функций		Г	—	2
45	Таймер		Г	—	0
46	Текстовый редактор типа «Блокнот»		Т/Г	—	3
47	Файловый менеджер (упрощенный FAR)		Т	—	3
48	Электронная таблица (упрощенный Excel)		Т/Г	—	3
49	Электронные часы		Г	—	0
50	Эмулятор механических часов – будильник		Г	—	0
51	Эмулятор механических часов (настенных)		Г	—	0
52	Генерация и прохождение лабиринта	[5 гл.8]	Г	О	2
53	Головоломка «Солитер»	[6 гл.16]	Г	О	1
54	Головоломка типа «15» (фишки разного размера)	[6 гл.33]	Г	О	3
55	Игра «Блэк»	[6 гл.30]	Г	ЧК	2
56	Игра «Большие гонки»	[7 с.106]	Г	2+Ч	3
57	Игра «Бридж-ит»	[6 гл.5]	Г	ЧК	3
58	Игра «Mastemind» (быки и коровы)	[5 гл.23]	Г	ЧК	2
59	Игра «Быки и коровы»	[7 с.176]	Г	2ЧС	3

Окончание таблицы 2

№	Темы курсовых работ	Ссылки	Текст/ гра- фика	Вид игры	Слож- ность
60	Игра «Калах»	[7 с.124]	Г	2Ч	2
61	Игра «Кант»	[7 с.114]	Г	2Ч	2
62	Игра «Квадратобоязнь» 7x7	[7 с.132]	Г	2Ч	2
63	Игра «Лиса и гуси»	[7 с.160]	Г	2Ч	1
64	Игра «Морской бой»	[7 с.170]	Г	2ЧС	2
65	Игра «Найаут»	[7 с.149]	Г	2Ч	2
66	Игра «Ползунок»	[7 с.77]	Г	2Ч	2
67	Игра «Реверси» (Отелло)	[7 с.38]	Г	2Ч	2
68	Игра «Тогуз-кумалак»	[7 с.124]	Г	2Ч	2
69	Игра «Тхаайям»	[7 с.141]	Г	4Ч	2
70	Игра «Гонки»	[8 гл.33]	Г	2Ч	2
71	Игра Леутуэйта	[9 гл. 12]	Г	ЧК	2
72	Игра «Менеджмент»	[5 гл.6]	Т/Г	2+ЧС	3
73	Игра «Ним»	[10 гл.14]	Г	ЧК	1
74	Калькулятор для программиста (целые числа, +, -, /, *, , &. ~. ^, перевод в 2,8,10,16 с.с.)		Т/Г	—	2
75	Колония жизни	[5 гл.2]	Г	—	2
76	Моделирование движения на автостраде	[5 гл.17]	Г	—	2
77	Программа для просмотра файла в шестнадцатиричном виде (как в FAR Просмотр/Код)		Т	—	2
78	Программа для решения пасьянса	[5 гл.19]	—	—	3
79	Программа для составления кроссворда	[5 гл.7]	Г	—	3
80	Самообучающаяся машина для игры в 6 шашек	[6 гл.14]	Г	ЧК	2
81	Утилита для архивирования и разархивирования текста	[5 гл.11]	—	—	3
82	Утилита для поиска файлов (командная строка, аналог Поиск файла в FAR)		—	—	2

3 ПОСТАНОВКА ЗАДАЧИ

На первом этапе разработки необходимо описать предметную область, решаемую задачу, все требования, допущения и ограничения. Постановка задачи выполняется на естественном языке. При необходимости для пояснений используются графические изображения. При оформлении постановки задачи по некоторому шаблону получившийся документ может называться

спецификация, документ-концепция или устав проекта в зависимости от включаемых в него разделов.

Целью постановки задачи является согласование с заказчиком (в курсовой работе — это руководитель) выполняемых программой функций, а также показать, что разработчик правильно понял задачу, учитывает все важные для заказчика требования и ограничения.

Для игры источником требований является в первую очередь правила игры, поэтому они включаются в постановку задачи. Например, существует несколько модификаций пасьянса Косынка, и в постановке задачи фиксируется ваш выбор варианта. Способ ввода (мышь и/или клавиатура) согласуется с заказчиком и становится требованием после согласования. Аналогичным образом разработчиком предлагается функциональность (feature) программы и эскизы (wireframe) интерфейса, которые после согласования превращаются в требования (рисунок 1 в приложении И).

Примеры требований к функциональности можно найти в приложениях И и Ж.

Использование определенной библиотеки (winBGI или conio) и языка программирования C в данной курсовой работе является ограничением и поэтому включается в постановку задачи.

Выбор среды разработки (MinIDE, Visual Code, CodeBlocks) или версии компилятора (GNU C 9.2, Microsoft Visual C 2017) является допущением и в постановку задачи обычно не включается, так как во время разработки это предположение может измениться.

К требованиям относится содержание главного меню программы (новая игра, правила, о программе, выход), но конкретный внешний вид является допущением, поэтому интерфейс главного меню в постановке задачи можно представить схематически или просто ограничиться перечислением пунктов.

Критерии оценки этапа для БРС:

– сформулирована тема, используемый язык программирования и библиотеки – 1 балл;

– указаны все функциональные требования (выполняемые функции, правила игры) – 2 балла;

– описан интерфейс (пункты меню, кнопки, используемые устройства взаимодействия) – 2 балла.

Итого максимум 5 баллов.

4 РАЗРАБОТКА АЛГОРИТМА

Алгоритмы + структуры данных = программы.

Н. Вирт

Прежде чем выполнять какой-то алгоритм, нужно иметь объекты, к которым применяются эти операции. Поэтому объектно-ориентированный подход требует на этом этапе выявить сущности (объекты) в предметной области и определить их свойства. С другой стороны, алгоритмы могут требовать данные определенной структуры (например, хранение данных в дереве для быстрого поиска вместо списка). Также на этом этапе строится математическая модель –

система математических соотношений (формул, уравнений, неравенств), отражающих существенные связи между объектами или явлениями, например, связь между ячейками матрицы и её представлением на экране или выигрышная стратегия компьютера.

Для простых программ достаточно базовых типов данных (целых и вещественных чисел), одномерных и двумерных массивов, структур и массивов структур. Основные сущности предметной области следует представлять глобальными переменными, структурами и массивами. Явным образом передавать такие сущности через аргументы нужно только в том случае, если функция может применяться к нескольким похожим объектам. Временные сущности, необходимые только на время выполнения некоторого действия, определяются как локальные переменные и массивы внутри функции. Если их необходимо использовать в других функциях, то они передаются через аргументы (по значению или через указатель).

Каждой сущности нужно задать идентификатор, чтобы ссылаться на эти сущности в алгоритмах. Пример описания основных сущностей для графического редактора можно найти в приложении И.

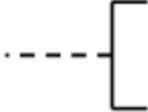
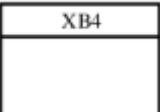
Для описания алгоритмов можно использовать запись на естественном языке, с помощью графических обозначений, на псевдокоде и непосредственно на языке программирования.

На начальных этапах разработки алгоритма рекомендуется использовать графические обозначения. Это позволяет лучше увидеть связи между основными частями алгоритма. При последующей детализации рекомендуется сначала переписать алгоритм на псевдокод. Псевдокод — это код на выбранном для реализации языке программирования с включениями на естественном языке там, где сложно записать действие как один оператор на языке программирования. В дальнейшем эти включения заменяются на несколько операторов или вызовы подпрограмм с сохранением описаний на естественном языке в форме комментариев. Таким образом сразу будет получаться код с необходимым и достаточным количеством комментариев.

Для графического представления алгоритма нужно использовать стандарт ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения», который является переводом международного стандарта ISO 5807:1985. Используемые символы указаны в таблице 3.

Необходимо нарисовать от 2 до 4 схем алгоритмов. Обязательной является схема основного алгоритма (основное меню программы). Так как разработка алгоритма происходит до написания кода, внутри символов рекомендуется использовать естественный язык и математические формулы, а не код на языке программирования С.

Таблица 3 — Символы, используемые в схемах программ

Символ	Название	Назначение
	Процесс	Символ отображает функцию обработки данных любого вида
	Данные	Символ отображает данные, носитель данных не определен
	Предопределённый процесс	Символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в подпрограмме (модуле)
	Решение	Символ отображает функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа. Соответствующие результаты вычисления могут быть записаны рядом с линиями, отображающими эти пути.
	Границы цикла	Символ, состоящий из двух частей, отображает начало и конец цикла. Обе части символа имеют один и тот же идентификатор. Условия для инициализации, приращения, завершения и т. д. помещаются внутри символа в начале или в конце в зависимости от расположения операции, проверяющей условие.
	Терминатор	Символ отображает вход из внешней среды и выход во внешнюю среду (начало или конец схемы программы)
	Соединитель	Символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать один и тот же уникальный идентификатор.
	Комментарий	Символ используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обводить группу символов.
	Символ с полосой	Символ указывает, что в этом же комплекте документации есть более подробное представление данной части алгоритма. В терминаторах такого подробного представления нужно указать идентификатор символа с полосой (рисунок 1а).

Не требуется точное соответствие между количеством символов в алгоритме и операторов в программе на языке С — один символ может представлять от 1 до 10 строк кода на языке С. Также структура алгоритма на этом этапе может отличаться от разделения операторов по функциям на этапе реализации программы, так как могут появиться новые обстоятельства, например, удаление дублирования кода или соблюдение рекомендаций по длине функции.

Критерии оценки этапа для БРС:

– есть описание структур данных и/или математическая модель для решаемой задачи – 5 баллов;

– есть по крайней мере 2 схемы алгоритмов, описывающих решение задачи – 5 баллов;

– схемы выполнены по ГОСТ 19.701–90 – 10 баллов.

Итого максимум 20 баллов.

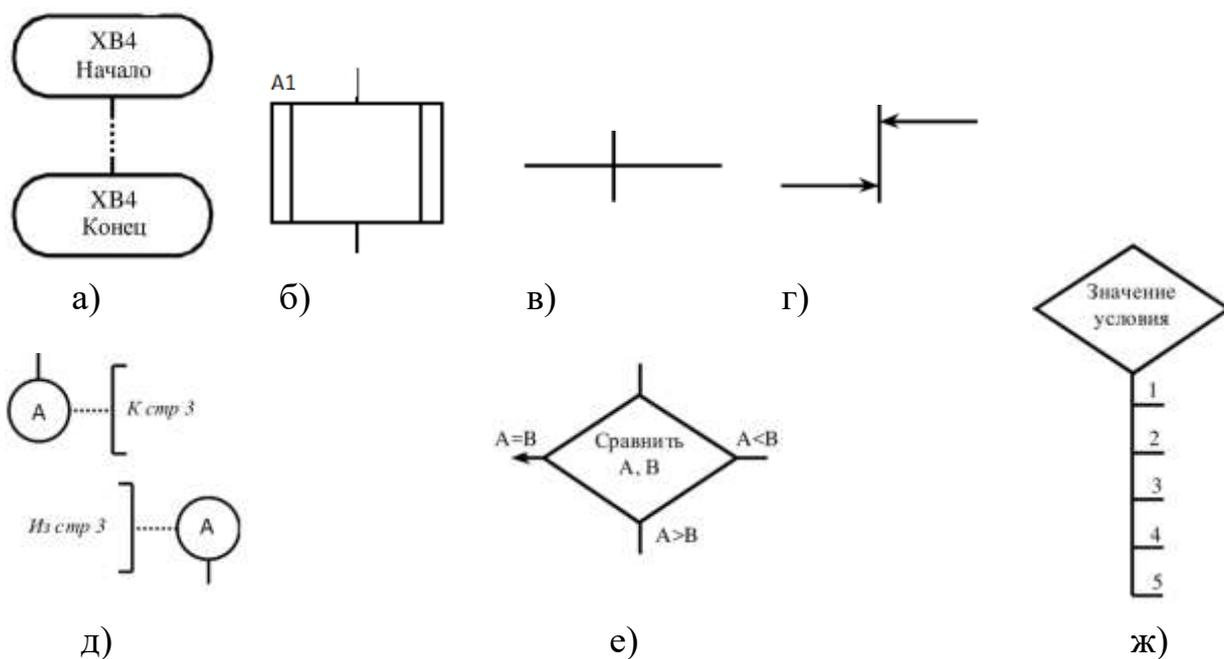


Рисунок 1 — Использование идентификаторов и линий в схемах алгоритмов

При рисовании символов рекомендуется делать их фиксированной ширины и располагать равномерно. Если необходимо сослаться на отдельные символы из других документов, то такие символы нужно снабжать идентификаторами, расположенными слева над символом (рисунок 1б).

Потоки управления в схемах показываются линиями. Линии в схемах должны подходить к символу либо слева, либо сверху, а исходить либо справа, либо снизу. Такие направления входа и выхода потока управления считаются стандартными. Линии должны быть направлены к центру символа. Если поток управления имеет направление, отличное от стандартного, стрелки должны указывать это направление. Также стрелки используются в случаях, когда необходимо внести большую ясность в схему (например, при соединениях линий). Если две или более линии объединяются в одну линию, место объединения должно быть смещено (рисунок 1г).

В схемах следует избегать пересечения линий. Пересекающиеся линии не имеют логической связи между собой, поэтому изменения направления в точках пересечения не допускаются.

Для минимизации излишних пересечений или слишком длинных линий в схемах линии рекомендуется разрывать и на концах разрыва использовать соединители с одинаковым идентификатором.

Если схема состоит из нескольких страниц, то для соединителей на разных страницах рекомендуется указать ссылки к страницам с помощью символа комментария (рисунок 1д).

Несколько выходов из символа следует показывать:

1) несколькими линиями от данного символа к другим символам (рисунок 1е);

2) одной линией от данного символа, которая затем разветвляется в соответствующее число линий (рисунок 1ж).

Примеры схем алгоритмов показаны на рисунках 2 и 3.

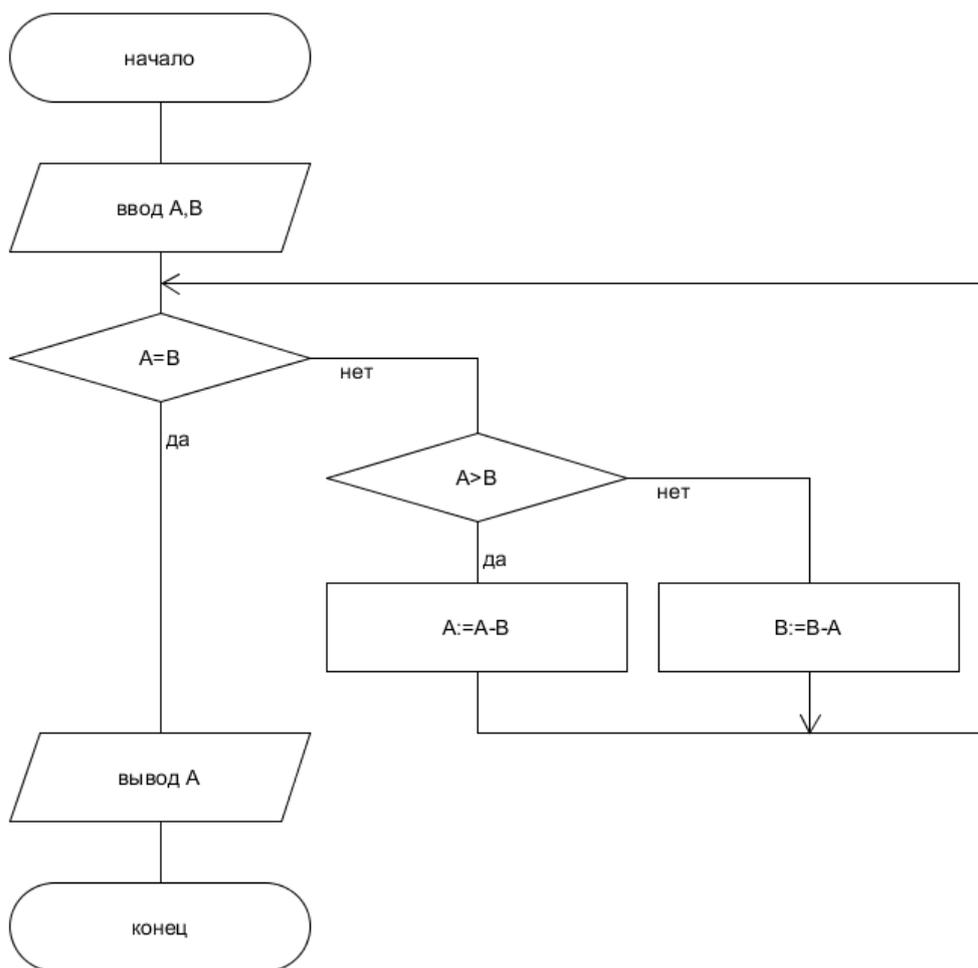


Рисунок 2 — Алгоритм Эвклида для нахождения НОД

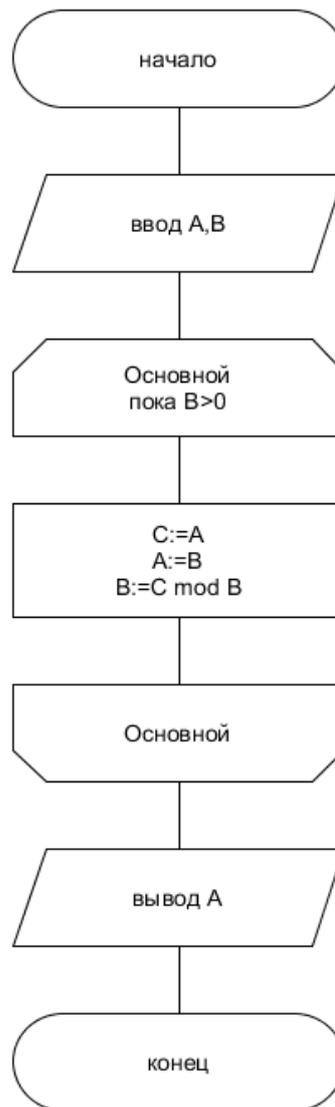


Рисунок 3 — Улучшенный алгоритм Эвклида и символы цикла

Для рисования схем используется специальная версия UMLet в среде MinIDE. Большинство online-инструментов использует либо предыдущий стандарт 1973 года, либо некоторую комбинацию с другими графическими нотациями (SDL, workflow, диаграмма деятельности UML). В большинстве инструментов отсутствует символ окончания цикла (хотя в помощи указывается, что он должен получаться путем переворачивания).

Выбор символов в UMLet выполняется из палитры «Flowchart GOST» двойным щелчком. Текст в символе изменяется в окне редактора. Аналогично из палитры можно выбирать подходящую линию. У линии можно указать характеристики тип стрелки и вид (сплошная или пунктирная) после «lt=», метки в начале и конце линии после «m1=», «m2=». Как установить другие характеристики, можно узнать с помощью Ctrl-пробел в окне редактора.

5 РЕАЛИЗАЦИЯ ПРОГРАММЫ

Программы должны составляться таким образом, чтобы их могли прочитать в первую очередь люди, а не машины. Программа — это документ для последующего использования.

Должен быть комментарий к каждой функции, глобальной переменной или структуре данных, объясняющий её назначение. Пояснительными комментариями нужно сопровождать те части программы, которые трудно понять без комментариев. Такой комментарий должен не переводить код на языке C на естественный язык, а содержать дополнительную информацию:

```
if(d<0) /* корней нет */
```

В среднем на 10 строк программы приходится один такой комментарий. Комментарии к каждой строке кода могут указывать на чужое авторство программы. Вы должны уметь объяснять действия в программе, основываясь только на коде, именах функций и переменных, а также комментариях к ним.

Логические блоки кода от 5 до 20 строк рекомендуется отделять пустыми строками, для такого блока можно сделать поясняющий комментарий, который поможет понять код, не выполняя анализ каждого оператора в блоке. Такой блок может соответствовать одному символу в схеме алгоритма.

Также не забывайте об отступах, показывающих структуру программы. Программа должна быть приятна для глаза. В состав MinIDE включена утилита для форматирования кода программы с отступами.

Правильный выбор имен — это залог удобочитаемости программ. Имена функций должны быть связаны с глаголами или отглагольными существительными. Для имен можно использовать сокращения от слов английского языка или русского языка, записанных транслитом. Глобальные имена должны быть достаточно длинными, имена локальных переменных можно делать короткими — от 1 до 3 символов. Имена констант нужно писать полностью прописными буквами.

Желательно, чтобы длина функции не превышала 50 строк, если получается больше — нужно выделить вспомогательные функции.

При написании кода нужно обращать внимание на повторяющиеся части кода. Рекомендуется выделить функцию или написать цикл по массиву данных, которыми эти части различаются. Вместо массива можно попытаться вычислить различающиеся данные по формуле.

Кроме повторения кода следует обращать внимание на повторение числовых и строковых констант. Таким константам нужно дать имена с помощью `#define` или `enum`. Рекомендуется использовать символьные константы вместо ASCII-кодов символов, например, '0' вместо 48.

В простейшем случае весь код программы можно записать в одном файле. Если количество строк кода в программе превышает 200-400, то можно разбить программу на несколько модулей. В заголовочном файле записываются заголовки функций, объявления структур и глобальных переменных. Заголовочный файл подключается ко всем модулям, которые используют эти

функции и переменные, а также к самому модулю для контроля корректности объявлений. Рассмотрим пример разбиения программы на модули.

Файл *module1.h*:

```
extern int size;  
double square(double x);
```

Файл *module1.c*:

```
#include "module1.h"  
int size=100;  
double square(double x)  
{ return x*x;  
}
```

Файл *module2.c*:

```
#include "module1.h"  
int main()  
{ double a=square(2.5);  
  ++size;  
}
```

В файле локальных настроек для программы, состоящей из нескольких модулей, в MinIDE нужно добавить строку:

```
build.goal=all
```

Результатом данного этапа является предварительная версия программы (альфа-версия). Проверяется в первую очередь понятность и качество кода, соответствие между реализованными функциями и требованиями в постановке задачи.

Руководителем могут быть указаны пропущенные требования, которые не вошли в постановку задачи или не были точно специфицированы, но текущая их реализация делает интерфейс неудобным для использования. Также руководителем могут быть выявлены ошибки во внешнем поведении программы, не соответствующие поставленной задаче. Эти замечания не влияют на оценку за данный этап, но должны быть исправлены на следующем этапе.

Критерии оценки этапа для БРС:

– понятность: есть комментарии ко всем функциям и глобальным переменным, используются именованные константы – 5 баллов;

– качество: нет дублирования кода, слишком длинных функций, нет ошибок в применении конструкций языка C, нет необоснованных усложнений – 5 баллов;

– полнота реализации требований – 5 баллов.

Итого максимум 15 баллов.

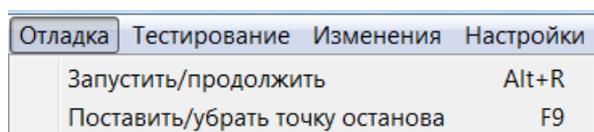
6 ТЕСТИРОВАНИЕ ПРОГРАММЫ И ИСПРАВЛЕНИЕ ОШИБОК

На данном этапе сначала производится изменение программы по результатам предыдущего этапа: исправляются ошибки в применении конструкций языка C, использования памяти, добавляются вспомогательные функции для уменьшения дублирования кода или слишком длинных функций, упрощается излишне усложненный код, изменяются комментарии или имена по

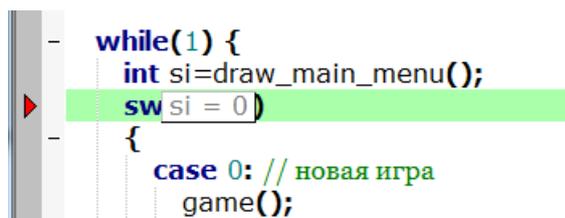
замечаниям руководителя, добавляются отсутствующие функции и улучшается пользовательский интерфейс по рекомендациям руководителя.

Затем производится тестирование программы. Тестирование — это процесс исполнения программы с целью обнаружения ошибок. Для этого рассматриваются несколько сценариев работы пользователя, и вы должны получить все возможные результаты программы, все сообщения об ошибочных действиях пользователя, если такие сообщения предусмотрены в вашей программе.

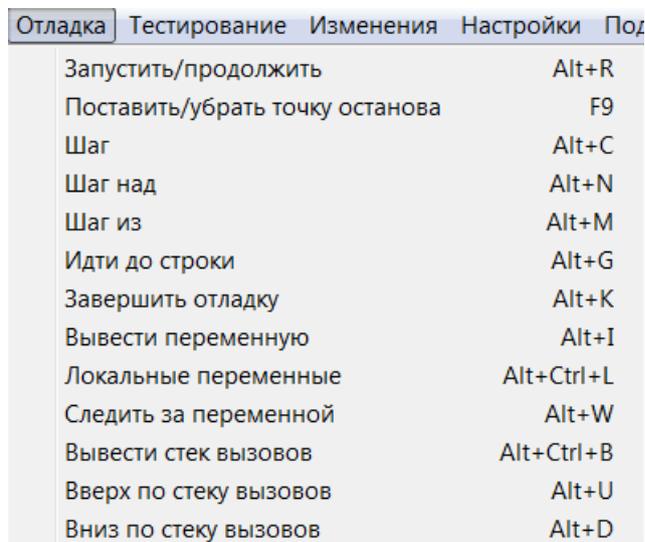
Если поведение программы не соответствует ожидаемым результатам, то выполняется процесс отладки — поиск части кода с ошибкой и её исправление. Для этого в среде MinIDE можно использовать отладчик. Сначала устанавливается точка останова на строке, непосредственно следующей за той строкой, результаты выполнения которой нужно проанализировать, за с помощью пункта «Поставить точку останова» из меню «Отладка» (рисунок 4а). После запуска программы с помощью пункта меню «Запустить/продолжить» из меню «Отладка» можно посмотреть значения интересующих переменных, подводя к ним курсор (рисунок 4б). Во время выполнения отладки в меню «Отладка» добавляются новые пункты для пошагового выполнения программы, просмотра стека вызовов и состояния локальных переменных в функции (рисунок 4в). Досрочное завершения выполнения программы и отладки завершается с помощью пункта меню «Завершить отладку».



а)



б)



в)

Рисунок 4 — Работа с отладчиком в MinIDE

Критерии оценки этапа для БРС:

- исправлены все замечания по коду, сделанные руководителем на предыдущем и на этом этапе – 5 баллов;
 - исправлены замечания по отсутствовавшей функциональности, неудобному интерфейсу, выявленные на предыдущем этапе – 5 баллов;
 - проверка внешнего поведения программы не выявило ошибок – 10 баллов.
- Итого максимум 20 баллов.*

7 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Параметры страницы: верхнее поле – 20 мм, нижнее поле – 20 мм, левое поле – 25 мм, правое поле – 10 мм. Нумерация страниц работы должна быть сквозной. Первой страницей считается титульный лист. Титульный лист, задание, аннотацию и оглавление включают в общую нумерацию страниц работы, но номера страниц на этих листах не проставляют. Номера страниц размещают в нижнем колонтитуле с выравниванием номера посередине.

Основной текст курсовой работы должен быть набран шрифтом Times New Roman размером 14 пт с одинарным межстрочным интервалом. Отступ первой строки абзаца — 0.7 см. В тексте не допускаются висячие строки (неполные строки в начале страницы). Для листингов программ используется шрифт Courier New, отступ первой строки абзаца — 0 см. Если количество строк кода не превышает пяти, то используется размер шрифта 14 пт, более пяти строк — 12.5 пт (чтобы получалось примерно 50 строк на страницу).

Опечатки, описки и графические неточности, обнаруженные в процессе подготовки работы, а также в результате проверки её руководителем допускается исправлять подчисткой или закрашиванием корректирующим карандашом или лентой, с последующим нанесением на том же месте исправлений, близких к компьютерному формату, шариковой или гелиевой ручками черного цвета.

Заголовки структурных элементов (частей) пояснительной записки печатают прописными буквами посередине, а заголовки разделов – прописными буквами с абзацного отступа. На разделы можно делить только основной материал и приложения. Разделы должны иметь порядковую нумерацию 1, 2, 3 и т. д. Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела включает номер раздела, например, 1.1, 1.2 и т. д. Заголовки подразделов печатают с абзацного отступа с прописной буквы. После номера раздела и подраздела в их названии точка не ставятся. После заголовка раздела в конце страницы должно размещаться не менее трех строк текста.

Все иллюстрации в работе (эскизы, схемы, графики, фотографии) называются рисунками. Название рисунка состоит из слова «Рисунок», его номера (используется сквозная нумерация рисунков в пределах всей работы) и наименования, отделенного символами «пробел» и тире от цифрового обозначения. Наименование пишется под рисунком посередине строки и может включать расшифровку обозначений, использованных в рисунке. На все рисунки в тексте работы должны быть ссылки. Первая ссылка имеет вид «рисунок 1» или «... в соответствии с рисунком 1», а все последующие ссылки на этот рисунок должны иметь вид – «см. рисунок 1».

Таблицы нумеруют арабскими цифрами. Слово «Таблица» и ее номер помещают слева над таблицей без абзацного отступа, например, «Таблица 1». После номера можно написать наименование таблицы, отделяя его символами «пробел» и тире от цифрового обозначения. Если таблица не помещается на странице, её делят на части. Слово «Таблица», ее номер и заголовок (при его наличии) указывают над первой частью таблицы. Над другими частями пишут

слова, например, «Продолжение таблицы 1» с указанием ее номера, а на последней странице – «Окончание таблицы 1». Все продолжения и окончание таблицы начинаются с повторения шапки таблицы. Рисунки и таблицы располагаются вслед за первым упоминанием о них в тексте.

Каждое приложение следует начинать с новой страницы. Наверху посередине страницы указывается слово «ПРИЛОЖЕНИЕ» прописными буквами и дается его обозначение заглавными буквами русского алфавита, начиная с А. Строкой ниже записывается тематический заголовок приложения с прописной буквы. Текст каждого приложения, при необходимости, может быть разделен на разделы, которые нумеруют в пределах каждого приложения (А.1, А.2 и т.д.). Аналогично иллюстрации и таблицы в приложениях нумеруют в пределах каждого приложения, например – Рисунок А.3, Таблица Д.2.

8 СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Структурными элементами (частями) курсовой работы являются [11]:

- *титульный лист*;
- *задание на работу*;
- *аннотация*;
- *оглавление*;
- *введение*;
- обзор литературы и постановка задачи;
- *основной материал по специальной части*;
- *заключение*;
- *библиографический список*;
- приложения.

Обязательные структурные элементы выделены полужирным шрифтом. Пример *титульного листа* приведен в приложении А. Пример *задания на работу* приведен в приложении Б.

Аннотацию помещают в пояснительной записке после задания. Аннотация включает:

- характеристику основной темы;
- проблемы объекта (если есть);
- цели (и задачи) работы;
- результаты работы.

Пример оформления аннотации показан в приложении В.

Оглавление состоит из перечня частей, разделов и подразделов работы и включает: введение, наименование всех разделов и подразделов основного материала, заключение, библиографический список и наименование приложений, для каждого из которых указываются номер страниц, с которых начинаются эти элементы курсовой работы. От конца текста до номера страницы дается отточие.

Пример оформления оглавления показан в приложении Г.

Во *введении* должна быть раскрыта актуальность темы курсовой работы, приведены цель и задачи работы, объект и предмет работы, а также показана

практическая применимость полученных автором результатов. Шаблон введения показан в приложении Д.

Заключение должно содержать:

- краткие выводы по результатам выполнения курсовой работы и оценку полноты решений поставленных в работе задач и достижения цели работы;
- рекомендации по конкретному использованию результатов курсовой работы;
- оценку результативности или эффективности предлагаемого решения.

Пример оформления заключения показан в приложении Е.

Пример оформления библиографического списка показан в приложении Ж.

В **основной части** курсовой работы приводятся данные, отражающие сущность, методику и основные результаты выполненной работы. Основная часть должна содержать следующие разделы:

- постановка задачи;
- разработка алгоритма;
- руководство пользователя.

Постановка задачи включает краткое описание задачи и описание взаимодействия с пользователем.

Разработка алгоритма должно содержать:

- пояснения по алгоритму, структурам данных и математической модели;
- 2 или более схемы алгоритма.

Руководство пользователя содержит описание работы пользователя от запуска программы до выхода. Например, правила игры, смысл появляющихся сообщений, управляющие клавиши. Не должно быть: "интерфейс является интуитивно понятным и не нуждается в описаниях" или "как работать с программой написано в подсказке, появляющейся по клавише F1, поэтому здесь не описывается". Пример оформления руководства пользователя показан в приложении И.

В **приложения** включаются материалы, связанные с выполненной курсовой работой, которые по каким-либо причинам не были включены в основную часть. В приложение А включаются исходные тексты программы, разделенные на подразделы по файлам. Пример оформления приложения показан в приложении К.

На данном этапе оценивается оформление пояснительной записки и руководства пользователя, так как постановка задачи и разработка алгоритма были уже оценены на предыдущих этапах. При грубых нарушениях правил форматирования, изложенных в разделе 7, пояснительная записка возвращается на доработку.

Критерии оценки этапа для БРС:

- оформление пояснительной записки в соответствии с шаблоном и правилами форматирования – 6 баллов;
- руководство пользователя содержит скриншоты интерфейса – 2 балла;
- руководство пользователя содержит достаточно подробное описание процесса взаимодействия – 2 балла.

Итого максимум 10 баллов.

9 ЗАЩИТА КУРСОВОЙ РАБОТЫ

В последнюю неделю семестра проводится защита курсовой работы. Защита проводится перед комиссией, состоящей не менее чем из двух преподавателей.

На защиту студент предоставляет:

1. Задание на курсовую работу (оформляется в начале семестра).
2. Программный продукт.
3. Пояснительную записку, содержащую описание разработки, программную документацию и соответствующие иллюстрации. В приложении приводится исходный код программы.

На защите студент коротко (3-5 минут) докладывает об основных проектных решениях, принятых в процессе разработки, и отвечает на вопросы членов комиссии.

План презентации:

- титульный слайд;
- постановка задачи – 1 слайд (сократить при необходимости);
- разработка алгоритма – 2-3 слайда;
- особенности реализации (за что себя хочется похвалить) – 1-3 слайда;
- скриншоты интерфейса – 1-3 слайда;
- заключение – 1 слайд.

Каждый слайд, кроме титульного должен иметь номер в правом нижнем углу. Пример презентации представлен в приложении М.

Критерии оценки этапа для БРС:

- подготовлена презентация – 5 баллов;
- в устном докладе студент показывает знания о проектных решениях в курсовой работе, свободно оперирует терминами применительно к рассматриваемой задаче – 10 баллов;
- студент может ответить на вопросы членов комиссии – 15 баллов.

Итого максимум 30 баллов.

Оценка «отлично» выставляется за курсовую работу, которая полностью соответствует техническому заданию, работоспособна во всех режимах, пояснительная записка имеет логичное, последовательное изложение материала с соответствующими выводами и обоснованными положениями. При защите студент показывает глубокое знание вопросов темы, свободно оперирует данными разработки, легко отвечает на поставленные вопросы.

Оценка «хорошо» выставляется за курсовую работу, которая полностью соответствует техническому заданию, работоспособна в большинстве режимов, пояснительная записка имеет грамотное, последовательное изложение материала с соответствующими выводами, однако с не вполне обоснованными положениями. При защите студент показывает знание вопросов темы, свободно оперирует данными разработки, без особых затруднений отвечает на поставленные вопросы.

Оценка «удовлетворительно» выставляется за курсовую работу, которая не полностью соответствует техническому заданию, работоспособна только в части режимов, пояснительная записка базируется на теоретическом материале,

но имеет поверхностный анализ, есть непоследовательность в изложении материала, представлены необоснованные положения. При защите студент показывает слабое знание вопросов темы, неуверенность в оперировании данными разработки, не всегда дает аргументированные ответы на поставленные вопросы.

Оценка «неудовлетворительно» выставляется за курсовую работу, которая не соответствует техническому заданию, не работоспособна, пояснительная записка не отвечает требованиям, изложенным в методических рекомендациях, не содержит выводов и обоснованных положений. При защите студент показывает слабое знание вопросов темы, неуверенность в оперировании данными разработки, не всегда дает аргументированные ответы на поставленные вопросы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Темы для курсовых работ. – URL: <https://ipc.susu.ru/42822.html>.
- 2 URL: <http://www.lightst.ru/card/Games/vosmerki.htm>.
- 3 URL: <https://ru.wikipedia.org/wiki/Видеопокер>.
- 4 URL: https://ru.wikipedia.org/wiki/Покер_на_костях.
- 5 Уэзерелл, Ч. Этюды для программистов. – М. : Мир, 1982. – 287 с.
- 6 Гарднер, М. Математические досуги. – М.: Мир, 1972. – 496 с.
- 7 Гик, Е. Интеллектуальные игры и развлечения / Е. Гик, А. Сухарев. — М: ФАИР-ПРЕСС, 1999. — 464 с.
- 8 Гарднер, М. Математические новеллы. – М.: Мир, 1976. – 456 с.
- 9 Гарднер, М. Путешествие во времени. – М.: Мир, 1990. – 336 с.
- 10 Гарднер, М. Математические головоломки и развлечения. – М.: Мир, 1999. – 447 с.
- 11 СТО ЮУрГУ 21–2008 Стандарт организации. Система управления качеством образовательных процессов. Курсовая и выпускная квалификационная работа. Требования к содержанию и оформлению / составители: Т.И. Парубочая, Н.В. Сырейщикова, А.Е. Шевелев, Е.В. Шевелева. – Челябинск: Изд-во ЮУрГУ, 2008. – 55 с.

ПРИЛОЖЕНИЯ
ПРИЛОЖЕНИЕ А
ПРИМЕР ТИТУЛЬНОГО ЛИСТА КУРСОВОЙ РАБОТЫ

Министерство науки и высшего образования РФ
ФГАОУ ВО ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИУ)
Институт естественных и точных наук

Факультет математики, механики и компьютерных технологий
Кафедра прикладной математики и программирования

«Растровый графический редактор»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

по дисциплине «Языки программирования»

ЮУрГУ–01.03.02.2020.313.ПЗ КР

Руководитель,

_____ *Демидов А.К.*

« ___ » _____ 2020 г.

Автор работы:

Студент группы: ЕТ – 112

_____ *Иванов И.И.*

« ___ » _____ 2020 г.

Работа защищена с оценкой

_____ 2020 г.

Челябинск – 2020

ПРИЛОЖЕНИЕ Б
ПРИМЕР ЗАДАНИЯ НА КУРСОВУЮ РАБОТУ

Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Кафедра «Прикладная математика и программирование»
Направление Прикладная математика и информатика

УТВЕРЖДАЮ

Заведующий кафедрой ПМиП

_____ А.А.Замышляева

_____ 2020 г.

ЗАДАНИЕ

на курсовую работу студента

_____ Иванова И.И. _____

Группа ЕТ-112

1. Дисциплина Языки программирования
2. Тема работы Растровый графический редактор

3. Срок сдачи студентом законченной работы 1 июня 2020 г.

4. Перечень вопросов, подлежащих разработке

- 1) постановка задачи и алгоритм программы;
- 2) реализация программы на языке С
- 3) оформление программной документации (руководство пользователя, листинг кода) и отчета по курсовой работе
- 4) презентация проектных решений для защиты КР (постановка задачи, разработка алгоритма, особенности реализации)

5. Календарный план

Наименование разделов (этапов) курсовой работы	Срок выполнения разделов (этапов) работы	Отметка о выполнении руководителя
постановка задачи	18.02.2020-03.03.2020	
разработка алгоритма	04.03.2020-24.03.2020	
реализация программы	25.03.2020-21.04.2020	
тестирование программы, улучшение и исправление ошибок	22.04.2020-10.05.2020	
оформление программной документации и отчета по курсовой работе	11.05.2020-20.05.2020	
защита курсовой работы	21.05.2020-01.06.2020	

Руководитель работы _____ / _____

Студент _____ / _____

(подпись)

(расшифровка)

ПРИЛОЖЕНИЕ В ПРИМЕР ОФОРМЛЕНИЯ АННОТАЦИИ НА КУРСОВУЮ РАБОТУ

АННОТАЦИЯ

Иванов И.И. Растровый графический редактор. – Челябинск: ЮУрГУ, ЕТ-112, 2020. – 24 с., 5 ил., библиографический список – 3 наим., 1 прил.

В курсовой работе описывается разработка растрового графического редактора на языке программирования С.

Целью курсовой работы является получение навыков создания программного обеспечения на основе структурного подхода при использовании интегрированной среды разработки.

В процессе работы были выполнены все этапы разработки программы: постановка задачи, проектирование программы, включающее нисходящее проектирование и структурное программирование, реализация и отладка программы.

Пояснительная записка содержит результаты выполнения этих этапов, разработанные структуры данных и схемы алгоритмов, использованные математические модели, а также руководство пользователя программы, включающее примеры интерфейса.

В результате работы была разработана растровый графический редактор, код которого приводится в приложении.

ПРИЛОЖЕНИЕ Г
ПРИМЕР ОГЛАВЛЕНИЯ
ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1 ПОСТАНОВКА ЗАДАЧИ	6
2 РАЗРАБОТКА АЛГОРИТМА.....	7
3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	11
ЗАКЛЮЧЕНИЕ	14
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	15
ПРИЛОЖЕНИЕ А	16

ПРИЛОЖЕНИЕ Д ПРИМЕР ОФОРМЛЕНИЯ ВВЕДЕНИЯ

ВВЕДЕНИЕ

Актуальность темы. Четкое соблюдение порядка этапов разработки программного обеспечения, понимание целей каждого из этих этапов, грамотное применение технологий и сред разработки обеспечивает реализацию сложных программных систем качественно и в срок.

Цель работы – разработать растровый графический редактор

Задачи работы:

- научиться корректно осуществлять постановку задачи;
- изучить методики проектирования программ, включая нисходящее проектирование и структурное программирование;
- научиться разрабатывать и описывать алгоритмы на основе структурного подхода с применением метода пошаговой детализации и стандартных графических обозначений;
- научиться выполнять разработку и отладку программы для ее решения;
- получить навыки работы с различными средами программирования и прикладными библиотеками;
- овладеть способами планирования, организации, самоконтроля и самооценки деятельности;
- осуществлять целенаправленный поиск информации в сети Интернет по теории и практике программирования, математическим алгоритмам, библиотекам для разработки консольных и графических программ.

Объект работы – программа для рисования.

Предмет работы – применение технологий разработки программного обеспечения на основе структурного подхода и языка С для разработки программы.

Результаты работы можно использовать в процессе последующего обучения в соответствии с учебным планом подготовки бакалавров по направлению «Прикладная математика и информатика»

ПРИЛОЖЕНИЕ Е ПРИМЕР ОФОРМЛЕНИЯ ЗАКЛЮЧЕНИЯ РАБОТЫ

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были поставлены точные требования к программе, затем были выявлены элементы интерфейса пользователя, разработаны необходимые математические модели, определены и детализированы структуры данных и алгоритмы. После завершения проектирования алгоритмы были реализованы на языке С. Разработанный код был проверен на контрольных тестах и в код были внесены необходимые исправления. Для программы было разработано руководство пользователя. Таким образом, цель работы была достигнута, задачи – решены.

Результаты работы можно использовать в процессе последующего обучения в форме навыков практического применения структурного программирования для разработки сложных программных систем, понимания порядка этапов разработки программного обеспечения и достигаемых на каждом этапе результатов.

ПРИЛОЖЕНИЕ Ж
ПРИМЕР ОФОРМЛЕНИЯ БИБЛИОГРАФИЧЕСКОГО СПИСКА

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Комирев, А. Г. Языки C и Basic с примерами и упражнениями: учеб. пособие для 1 курса. — Челябинск : Издательский Центр ЮУрГУ , 2015. — 222с. — URL: http://www.lib.susu.ac.ru/ftd?base=SUSU_METHOD &key=000552716 (дата обращения: 01.04.2020).

2 Солдатенко, И. С. Практическое введение в язык программирования Си: учебное пособие / И. С. Солдатенко, И. В. Попов. — Санкт-Петербург : Лань, 2018. — 132 с. — URL: <https://e.lanbook.com/book/109619> (дата обращения: 01.04.2020).

3 Графическая библиотека WinBGIm. – URL: <https://ipc.susu.ru/20786.html> (дата обращения: 01.04.2020).

ПРИЛОЖЕНИЕ И

ПРИМЕР ОСНОВНОЙ ЧАСТИ ДЛЯ ГРАФИЧЕСКОГО РЕДАКТОРА

1 ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать программу, реализующую растровый графический редактор. Для разработки необходимо использовать язык программирования C и графическую библиотеку winBGIm.

Редактор должен иметь следующие возможности:

- рисование точек (свободное рисование);
- рисование линий;
- рисование прямоугольников (заполненных и нет);
- выбор цветов рисования и заполнения из 16 цветов;
- чтение и запись рисунка в формате BMP.

Примерный интерфейс программы показан на рисунке 1.

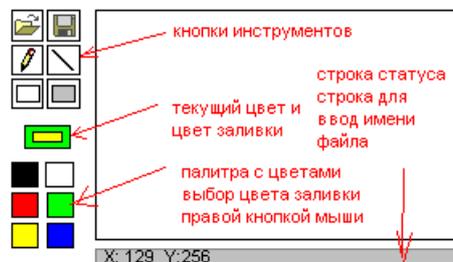


Рисунок 1 — Примерный интерфейс графического редактора

Выбор и выполнение действий выполняется с помощью мыши, ввод имени файла осуществляется с клавиатуры.

2 РАЗРАБОТКА АЛГОРИТМА

Основные сущности в программе:

- выбранный инструмент `tool` — целое число от 2 до 5;
- имя редактируемого файла `pic_name` — строка из 100 символов;
- текущий цвет `f_color` - целое число от 0 до 15;
- цвет заливки `b_color` - целое число от 0 до 15;
- массив из кнопок для инструментов `buttons`, каждая кнопка является структурой, содержащая координаты верхнего угла, размеры, имя файла с иконкой, загруженное изображение;

– строка статуса `status` — строка из 100 символов.

Координаты кнопок выбора цветов можно вычислить по номеру цвета (рисунок 2):

$$x_1 = 10 + \lfloor i/8 \rfloor \cdot 35$$

$$y_1 = 100 + (i \bmod 8) \cdot 35$$

$$x_2 = x_1 + 31$$

$$y_2 = y_1 + 31$$

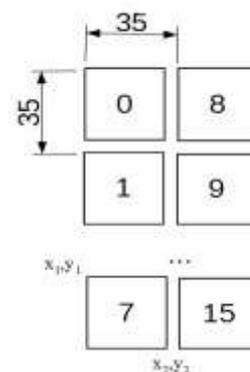


Рисунок 2 — Кнопки выбора цветов

Поэтому рисование этих кнопок и проверку нажатия можно проводить без определения вспомогательных структур данных.

Схема основного алгоритма редактора показана на рисунке 3. Алгоритм предварительного рисования линий и прямоугольников при нажатой кнопке мыши показан на рисунке 4. Если алгоритм возвращает 1, то происходит рисование линии или прямоугольника в соответствии с текущим цветом и цветом заливки.

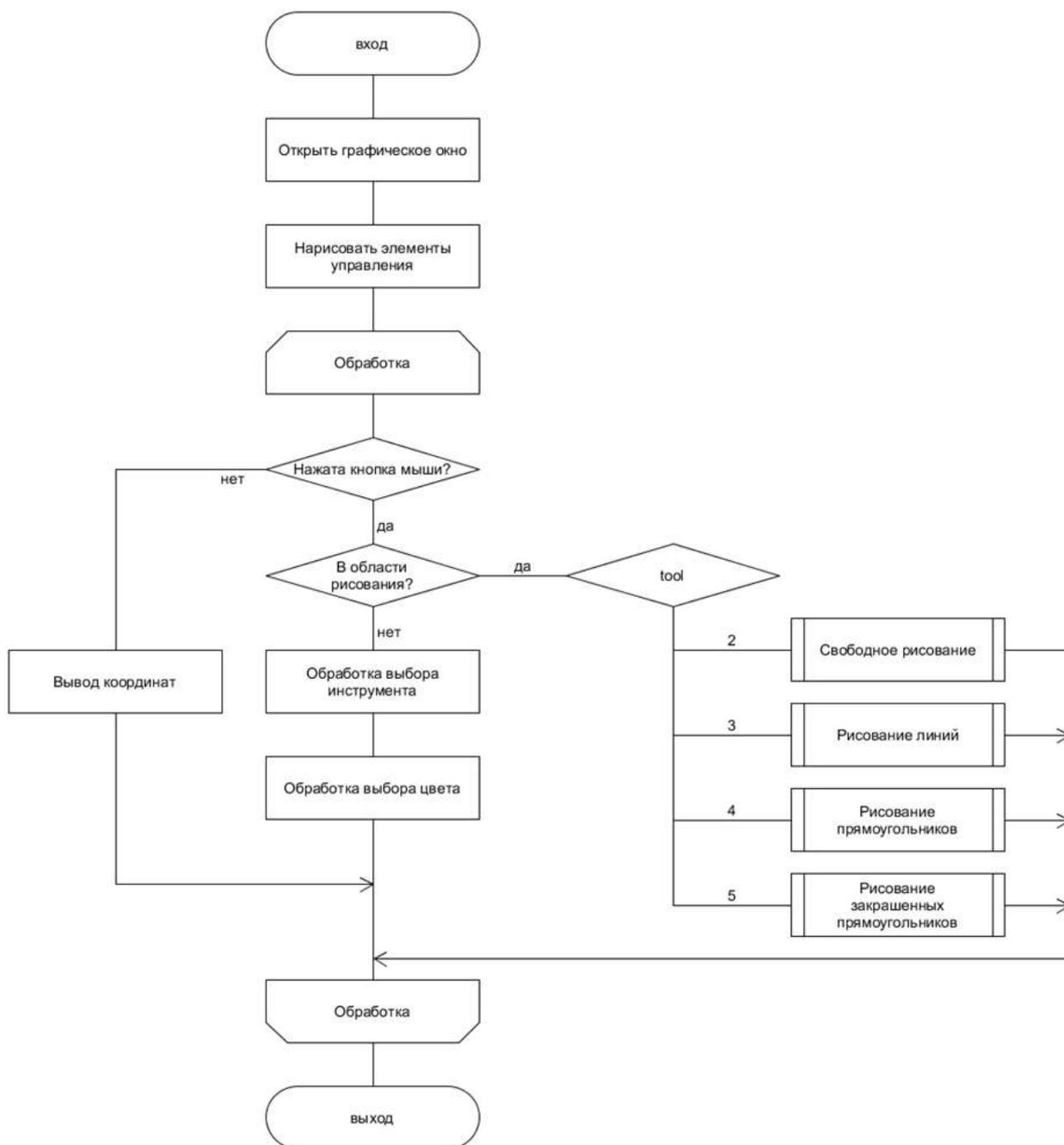


Рисунок 3 — Основной алгоритм программы

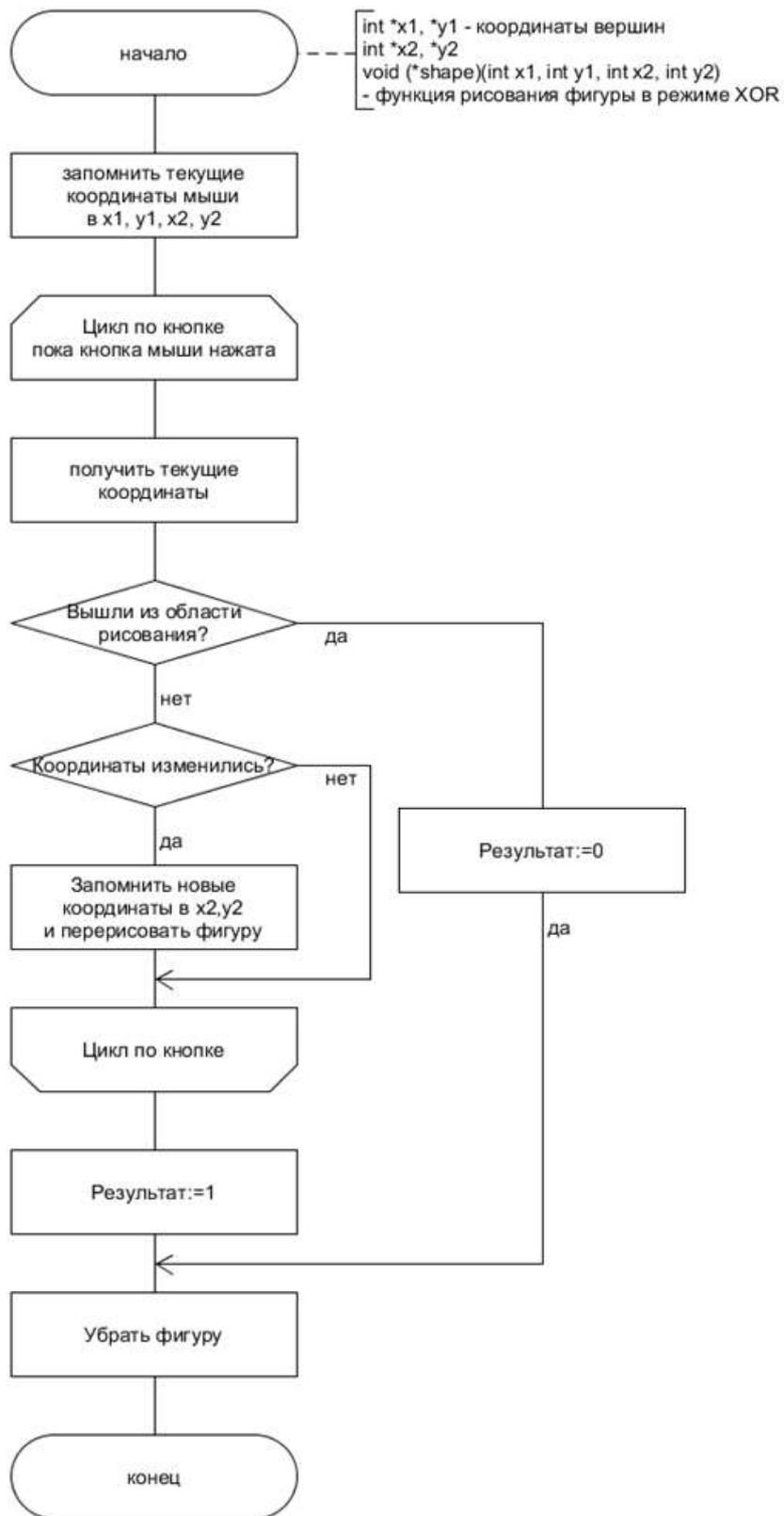


Рисунок 4 — Алгоритм предварительного рисования линий и прямоугольников

3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При запуске программы откроется окно редактора (рисунок 5).

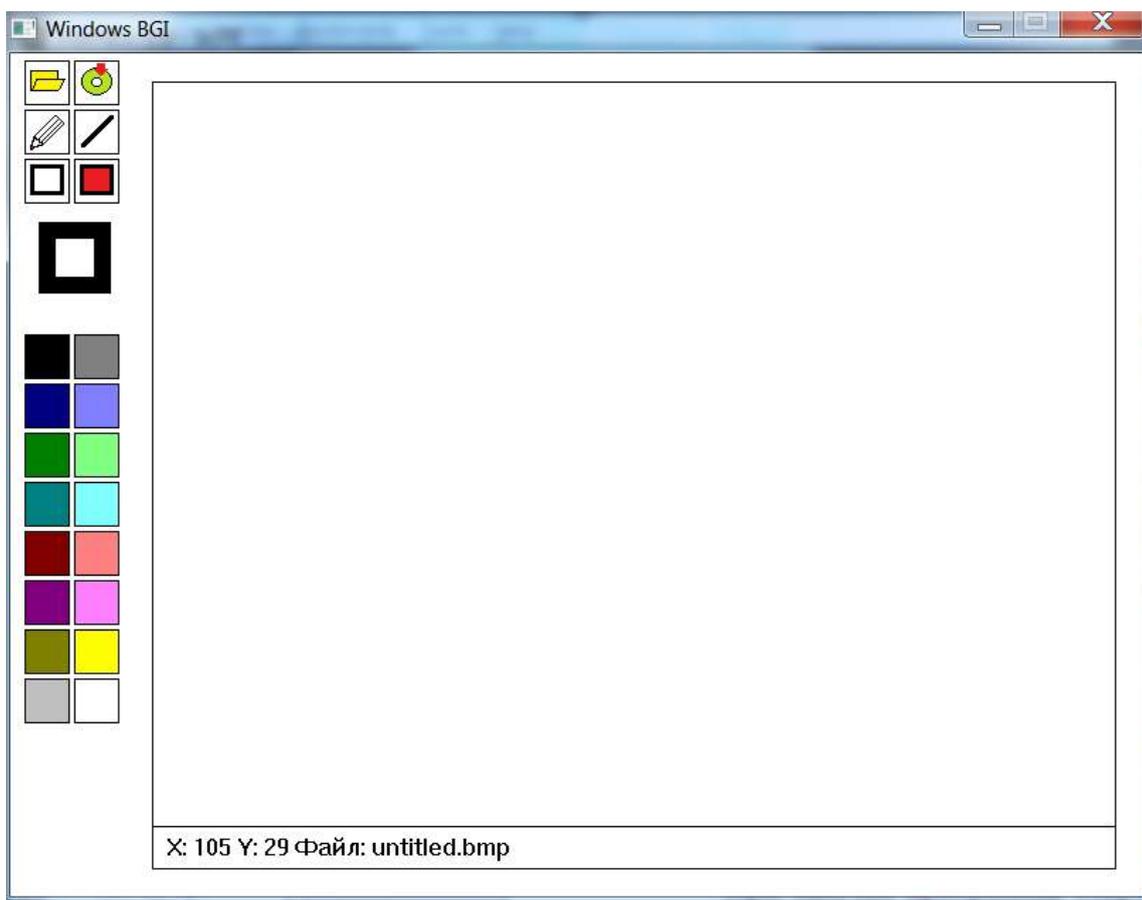


Рисунок 5 — Окно редактора

Белое поле в центре – поле для рисования. При перемещении мыши по полю в левом нижнем углу в строке состояния высвечиваются текущие координаты курсора в системе координат: ось X направлена из верхнего левого угла белого поля вправо, ось Y – вниз. При выходе за границы поля для рисования координаты не меняются.

Слева пользователь видит палитру из 16 цветов. Нажатием левой кнопки мыши выбирается цвет линий, правой – цвет заливки. Эти цвета отображаются в прямоугольнике над палитрой. При запуске программы цвет линий черный, цвет заливки — белый.

Рассмотрим по порядку кнопки инструментов слева направо сверху вниз (рисунок 6).



Рисунок 6 — Палитра инструментов

Первая кнопка в верхнем ряду — загрузка изображения. При нажатии на эту кнопку в строке состояния появится подсказка «Введите имя файла». По умолчанию имя файла «untitled.bmp». С помощью клавиш Backspace и алфавитно-цифровых клавиш можно изменить путь и имя файла. При нажатии Enter файл будет загружен. Если файл не существует, загрузка не производится.

Вторая кнопка в верхнем ряду — сохранение изображения. Аналогично, пользователь должен ввести путь и имя файла для сохранения и нажать Enter.

Для свободного рисования необходимо щелкнуть по первой кнопке во втором ряду. Рисование происходит при нажатой левой кнопки мыши на поле для рисования.

Для рисования линий необходимо щелкнуть по второй кнопке во втором ряду. При нажатии левой кнопки мыши в поля для рисования начинается рисование линии. При отпускании — линии фиксируется и рисуется выбранным цветом. Пока кнопка нажата, рисование линии происходит цветом, контрастным к цвету на изображении (на белом фоне линия рисуется черным цветом, на черном фоне — белым цветом).

Для рисования прямоугольников используются кнопки в третьем ряду. Первая кнопка — для рисования не закрашенных прямоугольников, вторая — для закрашенных. Аналогично, рисование прямоугольников начинается при нажатии левой кнопки мыши в поля для рисования. При отпускании — рисуется выбранный вид прямоугольника.

Для завершения работы с программой необходимо щелкнуть по кнопке с крестиком в верхнем левом углу.

ПРИЛОЖЕНИЕ К ПРИМЕР ПОСТАНОВКИ ЗАДАЧИ ДЛЯ ИГРЫ В СЛОВА

Необходимо разработать программу для игры в слова для двух человек по сети. Для разработки необходимо использовать язык программирования С и графическую библиотеку winBGIm.

Правила игры:

- Первый участник может назвать любое слово.

- Далее ходы выполняются по очереди, каждый участник должен назвать слово, начинающееся с буквы, которой заканчивается слово, названное предыдущим участником. Если слово заканчивается на ь, ь или ы, то выбирается предыдущая буква.

- Нельзя называть слово, которое уже было названо предыдущим участником.

Программа должна обеспечить проверку правил игры.

Главное меню программы содержит пункты «Новая игра», «Правила игры», «О программе», «Выход».

В процессе игры должен высвечиваться список использованных слов. Выбор и выполнение действий, ввод слов осуществляется с клавиатуры.

ПРИЛОЖЕНИЕ Л
ПРИМЕР ОФОРМЛЕНИЯ ПРИЛОЖЕНИЯ С ИСХОДНЫМ КОДОМ

ПРИЛОЖЕНИЕ А

А.1 Файл interface.h

```
#ifndef INTERFACE_H
#define INTERFACE_H

// область рисования
#define PAREA_X 100
#define PAREA_Y 20
#define PAREA_W 681
#define PAREA_H 531
// выбор цветов
#define COLORS_X 10
#define COLORS_Y 200
#define COLORS_D 35

extern int f_color, // цвет рисования
         b_color, // цвет закраски
         tool; // текущий инструмент
extern char pic_name[100]; // имя файла

void edit_name(); // ввести имя файла
void set_xy(int x, int y); // показать текущие координаты в
строке состояния
void draw_colors(); // нарисовать текущие цвета
int in_parea(int x, int y); // в области рисования?

#endif
```

А.2 Файл interface.cpp

```
#include <graphics.h>
#include <string.h>
#include <stdio.h>
#include "interface.h"

int f_color=BLACK, // цвет рисования
    b_color=WHITE, // цвет закраски
    tool=2; // инструмент карандаш
char pic_name[100]="untitled.bmp";

void draw_colors() {
    int x1=20,y1=120,x2=70, y2=170;
    setfillstyle(SOLID_FILL,f_color);
    bar(x1,y1,x2,y2);
    ...
}
```

ПРИЛОЖЕНИЕ М ПРИМЕР ПРЕЗЕНТАЦИИ

Министерство науки и высшего образования Российской Федерации
ФГАОУ ВО «Южно-Уральский государственный университет (НИУ)»
Институт естественных и точных наук
Кафедра прикладной математики и программирования

Курсовая работа по дисциплине «Языки программирования» **Растровый графический редактор**

Автор работы:
студент группы ЕТ-112
Иванов И.И.

Постановка задачи

Необходимо разработать растровый графический редактор со следующими возможностями:

- рисование точек (свободное рисование);
- рисование линий;
- рисование прямоугольников (заполненных и нет);
- выбор цветов рисования и заполнения из 16;
- чтение и запись рисунка в формате BMP.



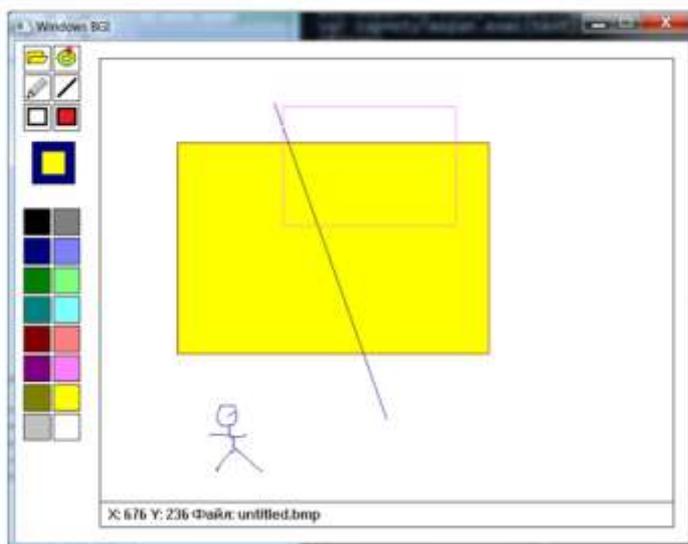
Разработка алгоритма

Основные сущности в программе:

- выбранный инструмент `tool` — целое число от 2 до 5;
- имя редактируемого файла `pic_name` — строка из 100 символов;
- текущий цвет `f_color` - целое число от 0 до 15;
- цвет заливки `b_color` - целое число от 0 до 15;
- массив из кнопок для инструментов `buttons`, каждая кнопка является структурой, содержащая координаты верхнего угла, размеры, имя файла с иконкой, загруженное изображение;
- строка статуса `status` — строка из 100 символов.



Пример работы программы



< 10 >

Заключение

В ходе выполнения курсовой работы были поставлены точные требования к программе, затем были выявлены элементы интерфейса пользователя, разработаны необходимые математические модели, определены и детализированы структуры данных и алгоритмы. После завершения проектирования алгоритмы были реализованы на языке С. Разработанный код был проверен на контрольных тестах и в код были внесены необходимые исправления. Для программы было разработано руководство пользователя. Таким образом, цель работы была достигнута, задачи – решены.

< 11

ОГЛАВЛЕНИЕ

1 ЭТАПЫ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ	3
2 ТЕМЫ ДЛЯ КУРСОВЫХ РАБОТ	3
3 ПОСТАНОВКА ЗАДАЧИ	6
4 РАЗРАБОТКА АЛГОРИТМА.....	7
5 РЕАЛИЗАЦИЯ ПРОГРАММЫ	13
6 ТЕСТИРОВАНИЕ ПРОГРАММЫ И ИСПРАВЛЕНИЕ ОШИБОК.....	14
7 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ	16
8 СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ.....	17
9 ЗАЩИТА КУРСОВОЙ РАБОТЫ.....	19
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	20
ПРИЛОЖЕНИЯ	
ПРИЛОЖЕНИЕ А	21
ПРИЛОЖЕНИЕ Б.....	22
ПРИЛОЖЕНИЕ В.....	23
ПРИЛОЖЕНИЕ Г	24
ПРИЛОЖЕНИЕ Д	25
ПРИЛОЖЕНИЕ Е.....	26
ПРИЛОЖЕНИЕ Ж.....	27
ПРИЛОЖЕНИЕ И	28
ПРИЛОЖЕНИЕ К.....	33
ПРИЛОЖЕНИЕ Л	34
ПРИЛОЖЕНИЕ М.....	35