МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»

ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

Етюды для программистов

Курсовая работа

По дисциплине «Введение в специальность»

студента 1 курса группы ПИ-201(2)

Садетов Алим Асанович

Направления подготовки 09.03.04 «Программная инженерия»

(код и наименование)

Научный руководитель

Профессор, д. ф.-м. н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Потапова Е. В.

(подпись, дата)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка)

Симферополь, 2021

# РЕФЕРАТ

**Название работы:** программирование вне в КФУ

**Проблема работы:** отсутствие практических заданий, приближенных к реальным задачам

**Актуальность работы**: почему лучше учить программирование самому

**Задача работы**: изучить информацию и преподнести ее людям.

# Оглавление

[РЕФЕРАТ 2](#_Toc67602936)

[Оглавление 3](#_Toc67602937)

[СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ 3](#_Toc67602938)

[Введение 5](#_Toc67602939)

[Глава 1 Что такое программирование 5](#_Toc67602940)

[1.1 Почему 5](#_Toc67602941)

[1.2 Что 5](#_Toc67602942)

[1.1.1 Когда 5](#_Toc67602943)

[Глава 2 ЖизнПочему с++ должен умереть 8](#_Toc67602944)

[2.1 Модель колонии жизни 8](#_Toc67602945)

[2.2 АЦПУ 9](#_Toc67602946)

[Заключение 11](#_Toc67602947)

[Список литературы 11](#_Toc67602948)

[Приложение 1 13](#_Toc67602949)

# СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

|  |  |
| --- | --- |
| ВУЗ | Высшее учебное заведение |
| ВКР | Выпускная квалификационная работа |
| ГАК | Государственная аттестационная комиссия |
| ГИА | Государственная итоговая аттестация |
| ГОСТ | Государственный стандарт |
| КФУ | Крымский федеральный университет |

# Введение

Программирование это круто но можно ли его выучить самому конечно можно для этого придётся правильно расставить приоритеты и возможно вообще забросить учёбу в вузе но это рискованно и можно отчислиться

# Что такое программирование

## 1.1 Почему

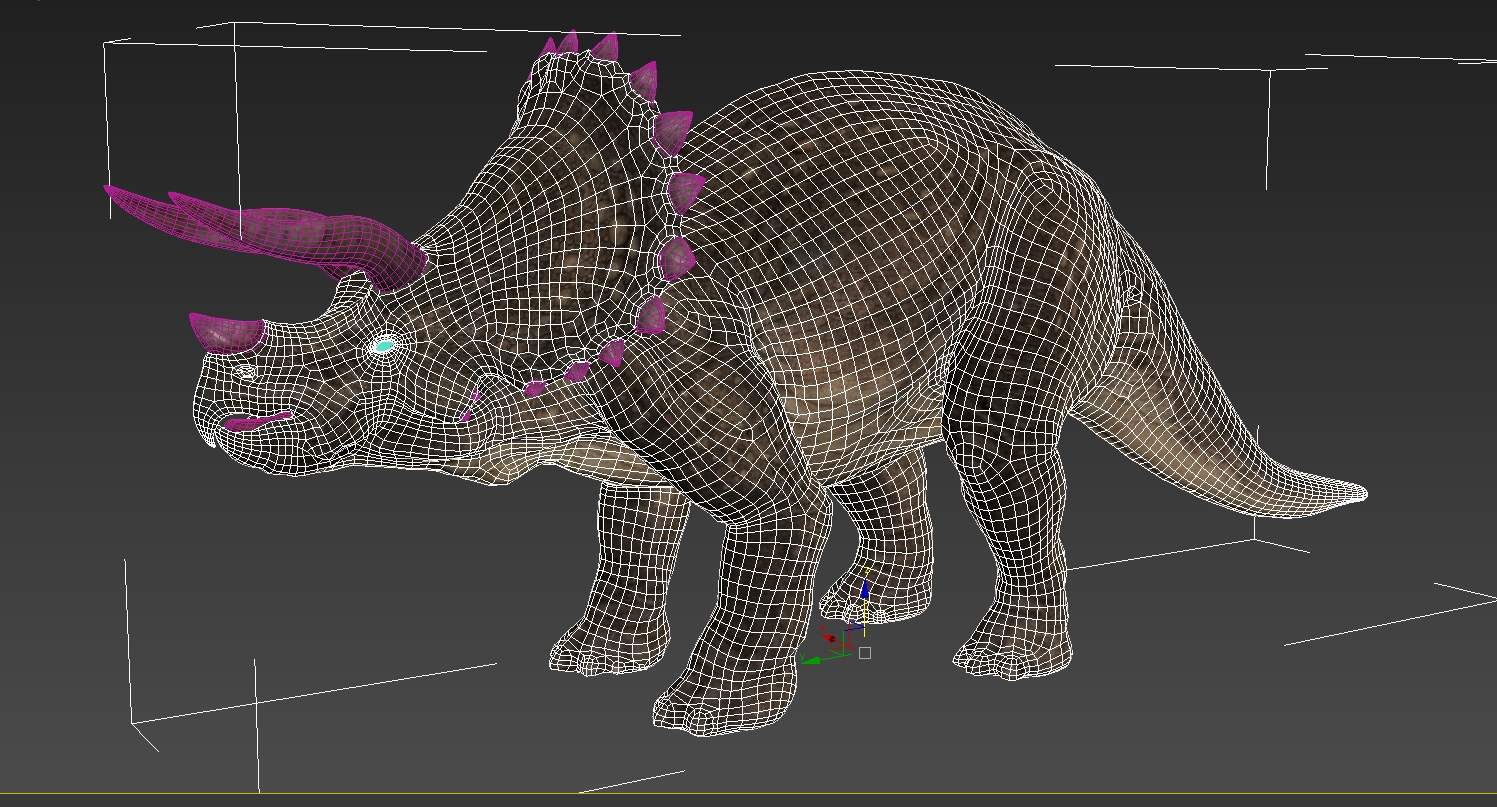
Здесь должен быть текст сейчас я его напишу на несколько страниц и так погнали текст поехали начинайся ну всё думаю хватит

## 1.2 Что

Конечно, результатом работы над этюдом должна быть понятная и четкая программа, стиль и комментарии которой соответствовали бы задаче и выбранному языку. Но этого мало. Еще необходим набор тестов, достаточный для демонстрации работы программы и ее реакции на экстремальные ситуации и неправильное обращение. Наряду с самой программой требуется краткое словесное описание методов решения. Особый упор в нем следует сделать на положенные в основу решения алгоритмы и структуры данных. Наряду с описанием программы программист должен с достаточной степенью правдоподобности хотя бы неформально проиллюстрировать ее правильность (при недостатке времени можно ограничиться рассмотрением ключевых мест). Наконец, должен быть произведен подсчет затраченных ресурсов, как людских, так и машинных; особое внимание следует обратить на обоснование затрат. Также следует указать, чему программист научился на примере этой задачи (на этот вопрос легко ответить, если сформулировать его в виде: «Что я в следующий раз сделаю иначе?»). Такой объем документации может показаться избыточным. Заметим, однако, что умению вовремя поставить точку тоже очень полезно научиться. Решение небольшой задачи не следует перегружать документацией. Один знакомый автору преподаватель определяет оценку на 40% тем, что студент убедил его в правильности программы, на 50% легкостью, с которой его удалось убедить, и только на 10% отличным программированием. Очень хорошая оценка — это 80% и более. А поскольку часть документации — результаты машинных прогонов, такая отметка означает, что программа произвела благоприятное впечатление и на преподавателя, и на ЭВМ.

### 1.1.1 Когда

Программирование появилось 200 лет назад и программировали не на компьютерах.

 Рисунок 1.1. тирекс

Science Citation Index. Institute for Scientific Information, Philadelphia, PA. Yearly.

Если вы хотите узнать побольше по какому-либо из затронутых в нашей книге направлений, можно воспользоваться цитированной литературой, затем — библиографией из этих работ и т. д. Но как найти работы, которые вышли в свет уже после перечисленных в книге? Если у вас есть некий источник по какой-либо теме, то в Science Citation Index можно найти работы, ссылающиеся на имеющуюся у вас. В каждом из ежегодных выпусков разъясняется, как им пользоваться, да и библиотекарь вам в этом поможет.

Конвей, Грис (Conway R., Gries D.). An Introduction to Programming, 2nd ed. Winthrop, Cambridge, MA, 1975.

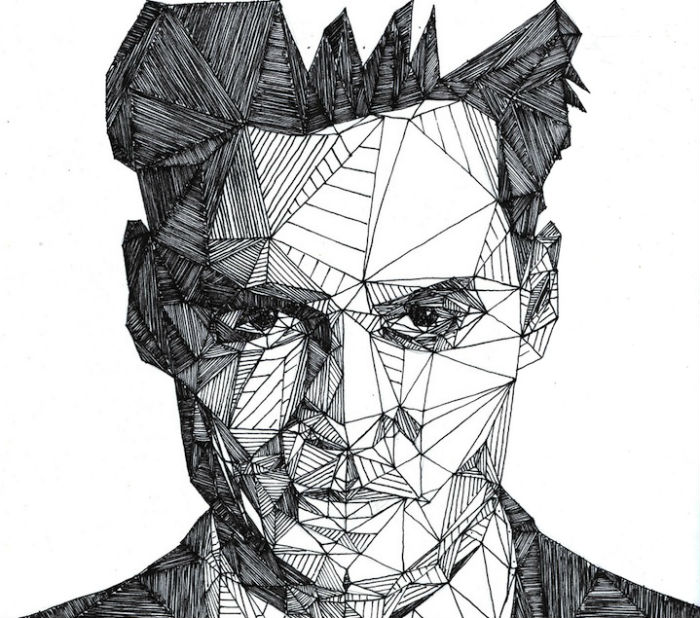


Рисунок 1.2. нуар

Строго говоря, это — введение в программирование (а заодно и хорошее руководство по PL/L). Но, кроме того, это прекрасный учебник по надежности и методам доказательства правильности программ. Перед тем как приступить к вашему первому этюду, имеет смысл повторить материал по построению программ, приведенный в этой книге.

Вирт (Wirth N.). Algorithms + Data Structures = Programs, Prentice-Hall, Englewood Cliffs, NJ, 1976.

Дейкстра (Dijkstra E. W.). A Discipline of Programming, Prentice-Hall, Englewood Cliffs, NJ, 1976. [Имеется перевод: Дейкстра Э. Дисциплина программирования.— М.: Мир, 1978.]

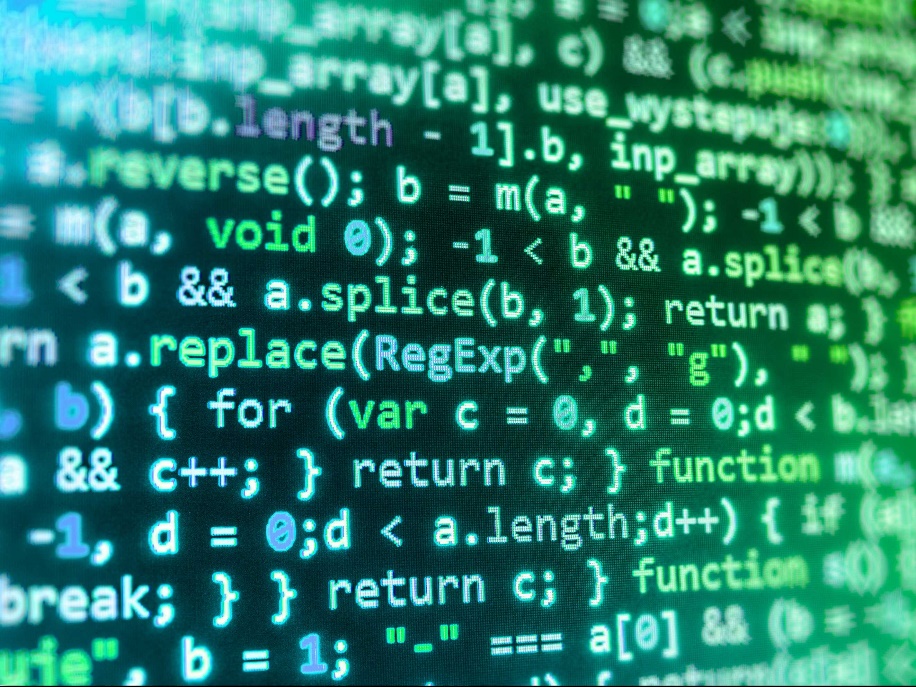


Рисунок 1.3. надеюсь с++ умрёт

# ЖизнПочему с++ должен умереть

## 2.1 Модель колонии жизни

С++ просто ужасен его уже ни один человек в мире не спасёт пожалуйста убейте его пока не поздно его не должны преподавать в вузах это просто ужасный язык.

|  |  |  |
| --- | --- | --- |
| № | Тема занятия | Количество  часов |
| Создание графических изображений с помощью Macromedia Flash | | |
| 1. | Введение: примеры использования Macromedia Flash технологии в сети Интернет; место Macromedia Flash среди Web-технологий; первое знакомство с программой Macromedia Flash; идея создания сайтов в Macromedia Flash; обсуждение работы с личными проектами; тематическое планирование интерфейса. | 1 |
| 2. | Интерфейс: рабочая область, ее параметры, сетка, линейка и магнит; полоса времени; ключевые кадры; слои; панель инструментов; инструменты; окна свойств; импорт изображений; работа с библиотекой объектов. | 1 |
| 3. | Рисование: линия, карандаш, ластик, кисть, круг, прямоугольник, текст, перо, пипетка, заливка линий и областей, выделения и деформация, лупа и рука, работа с кривыми, градиенты, текстуры. | 1 |

Таблица 2.1. Случайная таблица

Любая колония имеет преемника, но не у каждой есть предшественник. Такие изолированные колонии называются **садами Эдема**. Сад Эдема можно увидеть, только если поместить его на плоскость в качестве начальной конфигурации. Подумайте, как использовать вашу программу для нахождения сада Эдема.

Беркс (ред.) (Burks A. W. (Ed.)). Essays on Cellular Automata. University of Illinois Press, Urbana, IL, 1970.

Кодд (Codd E. F.). Cellular Automata. Academic Press, New York, NY, 1968.

Обе эти книги значительно серьезнее статей Гарднера в Scientific American. Вторая из названных книг познакомит вас с основами предмета, а книга Беркса представляет собой сборник разнородных статей, охватывающих всю область клеточных автоматов. После изучения этих книг читателю будет доступен практически весь математический материал.

Гарднер (Gardner Martin). Mathematical Games. Scientific American, 223, 10, pp. 120–123, October 1970, and 224, 2, pp. 112–117, February 1971. [Имеется перевод: Гарднер М. Математические досуги. — Мл Мир, 1972, с. 458.]

## 2.2 АЦПУ

Таким образом, гибнущая клетка может способствовать рождению новой, но рождающаяся клетка не может воскресить гибнущую, и гибель одной клетки, уменьшив локальную плотность населения, не может предотвратить гибель другой.

Так, например, колония ??? превращается в следующем поколении в ? а колония °° должно быть, живет неподалеку от райского Палм-Спрингс, поскольку она вообще никогда не меняется. На рис. 2.1 показана история еще одной колонии клеток Жизни.

Тема. Напишите программу, моделирующую колонию Жизни. Исходными данными служит начальное расположение клеток, а в качестве результата нужно получить вид сверху всех поколений колонии. Для вывода истории можно воспользоваться обычным устройством построчной печати (АЦПУ), но такой способ дает весьма неприглядные изображения. Если в вашем распоряжении имеется графопостроитель или графический терминал, воспользуйтесь их возможностями для получения более изящной картинки.

Рисунок 2.1. История одной колонии Жизни. Номер поколения выписан слева от каждой картинки. Найдите самостоятельно поколения 9 и 10.

(2.1)

(2.2)

где b – коэффициент при 2-ом члене многочлена

с – свободный член

Рекомендации исполнителю. Хотя этого и не видно из примеров, некоторые колонии разрастаются невероятным образом при весьма скромных начальных размерах. Есть другие колонии, которые медленно перемещаются по пустыне, переходя на все новые и новые территории. Ваша программа должна обрабатывать большие колонии без чрезмерной траты памяти или времени. Многократный просмотр большого массива для построения следующих поколений — это банальный подход; здесь программистская задача состоит в выборе более экономичных структур данных и алгоритмов. Вам, возможно, захочется испытать какой-либо метод, отслеживающий только занятые квадраты. Растущая или движущаяся колония может выйти из поля зрения, если его положение и границы зафиксированы, поэтому, вероятно, понадобится еще и метод вывода, перемещающий нашу точку зрения вслед за изменениями колонии.

Инструментовка. Для этой задачи подойдет язык APL благодаря наличию в нем операций над векторами и матрицами, однако можно использовать почти любой язык высокого уровня, если в нем предусмотрена работа с массивами. На примере этой задачи хорошо изучать, как сказывается использование языка ассемблера: насколько замедляется программирование и каков выигрыш в эффективности внутреннего цикла. Наконец, для тех, кто имеет доступ к оборудованию ЭВМ, интересным экспериментом могла бы быть микропрограммная реализация; машина при этом превращается в колонию Жизни.

# Заключение

В заключение могу только пожелать с++ скорейшей смерти

# Список литературы

1. Бородаев Д.В. Web-сайт как объект графического дизайна. Монография. — Х.: "Септима ЛТД", 2006. — С. 288.

2. Грибов Д.Е. Macromedia Flash 4. Интерактивная web-анимация. - М. ДМК. 2000. - 672 с.

3. Гончаров А.Ю. Web-дизайн: HTML, JavaScript и CSS. Карманный справочник. — "КУДИЦ-ПРЕСС", 2007. — С. 320.

4. Дунаев Владислав. Кухня Photoshop – СПб.: Симбол-плюс, 2006 г.

5. Кирсанов Д. Веб-дизайн: книга Дмитрия Кирсанова. – СПб.: Симбол-плюс, 1999 г.

# Приложение 1

Формулы С++ с комментариями, полиморфизм, наследование

#include<iostream>

#include<string>

using std::cout;

using std::string;

using std::endl;

/\*

\*полиморфизм

\*Виртуальные функции

\*virtual

\*override

\*Чистые виртуальные функции

\*Абстрактный класс

\* Виртуальный деструктор

\* Чисто виртуальный деструктор

\* Делегирующий конструктор (синтаксический сахар для удобства программиста)

\* метод базового класса в классе наследника

\* множестевнное наследование

\*/

class Car

{

public:

Car()

{

cout << "Вызван конструктор Car" << endl;

}

~Car()

{

cout << "Вызван деструструктор Car" << endl;

}

string str = "Поле класса машина";

void Drive()

{

cout << "Я еду!" << endl;

}

};

class Airplain

{

public:

~Airplain()

{

cout << "Вызван деструструктор Airplain" << endl;

}

Airplain()

{

cout << "Вызван конструктор Airplain" << endl;

}

string str1 = "Поле класса самолет";

void Fly()

{

cout << "Я лечу!" << endl;

}

};

class FlyingCar : public Car, public Airplain

{

public:

~FlyingCar()

{

cout << "Вызван деструструктор FlyingCAr" << endl;

}

FlyingCar()

{

cout << "Вызван конструктор FlyingCAr" << endl;

}

};

int main()

{

setlocale(LC\_ALL, "ru");

FlyingCar fl;

return 0;

}

# 