Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики и технологии программирования

Дисциплина: Программирование

К ЗАЩИТЕ ДОПУСТИТЬ

\_\_\_\_\_\_\_\_\_\_И. А. Удовин

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе  
на тему

Решение задач методом перебора

Студент

Руководитель

П. А. Бобрик

И. А. Удовин

МИНСК 2020

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики и технологии программирования

Дисциплина: Программирование

УТВЕРЖДАЮ

Заведующий кафедрой ИиТП

\_\_\_\_\_\_\_\_\_\_\_\_ Н. А. Волорова

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2020 г.

ЗАДАНИЕ

по курсовой работе студента  
Бобрика Павла Алексеевича

1. Тема работы: Решение задач методом перебора
2. Срок сдачи студентом законченной работы: 4 июня 2020 г.
3. Содержание пояснительной записки (перечень подлежащих разработке вопросов):

Введение. Основная часть. 1 Общие положения. 2 Алгоритм реализации проекта. 3.Выбор языка программирования . 3.1 Сравнение языков программирования. 4.Выбор среды разработки. 5 Организация и ход программы . 5.1 Аспекты для разработки праграммы. 5.2 Сравнение методов перебора 5.3 тестирование 6. Код программы. Заключение. Список использованных источников.

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсовой работы | Объём этапа, % | Срок выполнения этапа | Примечания |
| Разработка метода перебора | 80 | 30.05-31.05 |  |
| Разработка вывода доп.информации | 20 | 1.06 |  |

Дата выдачи задания: 06 марта 2020 г.

Руководитель И. В. Удовин

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ \_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ 5

ОСНОВНАЯ ЧАСТЬ 7

1 Общие положения 7

2 Алгоритм реализации проекта 7

3 Выбор языка программирования 8

3.1 Сравнение языков программирования 9

4 Выбор среды разработки. 9

5 Аспекты программного обеспечения 10

5.1 Функционал программы 10

5.2 Схема метода перебора 11

6 Тестирование 13

7 Код программы 13

ЗАКЛЮЧЕНИЕ 14

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15

**ВВЕДЕНИЕ**

В наше время информация играет важную роль в мире. Не менее важную роль играет защита информации, призванная «скрыть от ненужных глаз» важные для людей данные. Один из способов защиты информации - паролирование.

Паролирование сейчас применяется повсеместно. Чаще всего паролирование встречается в Интернете. Однако и в жизни часто встречаются пароли: например, банкомат требует ПИН-код, который тоже является паролем. Такой же код требуется ввести при включении телефона.

Пароль - это секретное слово или набор символов, предназначенный для подтверждения личности или полномочий. Пароли часто используются для защиты информации от несанкционированного доступа. В большинстве вычислительных систем комбинация «имя пользователя - пароль» используется для удостоверения пользователя. Паролирование - основной способ защиты информации.

Для повышения безопасности обычно используются следующие методы:

* Ограничение минимальной длины пароля.
* Требование повторного ввода пароля после определенного периода бездействия.
* Требование периодического изменения пароля.
* Назначение стойких паролей (генерируемых с использованием аппаратного источника случайных чисел, либо с использованием генератора псевдослучайных чисел, выход которого перерабатывается стойкими хэш-преобразованиями).

Исследования показывают, что около 40 % всех пользователей выбирают пароли, которые легко угадать автоматически. Легко угадываемые пароли (123, admin) считаются слабыми и уязвимыми. Пароли, которые очень трудно или невозможно угадать, считаются более стойкими.

Часто требуют следующие требования к паролям:

* Пароль не должен быть слишком коротким, поскольку это упрощает его взлом полным перебором. Наиболее распространенная минимальная длина - восемь символов. По той же причине он не должен состоять из одних цифр.
* Пароль не должен быть словарным словом или простым их сочетанием, это упрощает его подбор по словарю.
* Пароль не должен состоять только из общедоступной информации о пользователе.

Брутфорсом называется метод взлома учетных записей путем подбора паролей к ним. Термин образован от англоязычного словосочетания «brute force», означающего в переводе «грубая сила». Суть подхода заключается в последовательном автоматизированном переборе всех возможных комбинаций символов с целью рано или поздно найти правильную. С этой точки зрения поиск пароля можно рассматривать как математическую задачу, решение которой находится при достаточно большом количестве попыток. Программное обеспечение для брутфорса генерирует варианты паролей и проверяет каждый из них. С точки зрения математики решить задачу таким способом можно всегда, но временные затраты на поиски не во всех случаях оправдывают цель, так как поле поиска решений огромно.

Брутфорс — один из самых популярных методов взлома паролей к учетным записям онлайн-банков, платежных систем и других веб-сайтов. Впрочем, с ростом длины пароля этот метод становится неудобным, так как растет время, которое нужно на перебор всех вероятных вариантов. Также с его помощью можно проверять криптоустойчивость пароля.

Брутфорс еще называют методом исчерпывания, так как верная комбинация выявляется путем анализа всех возможных вариантов и отбрасывания каждого неподходящего сочетания.

**ОСНОВНАЯ ЧАСТЬ**

1. **Общие положения**
   1. Для разработки игры используется программа Visual Studio
   2. За концепцию взята основа программ BruteForce
   3. Для осуществления цели курсовой работы применяется язык программирования C#.
   4. В ходе написания игрового кода изучены некоторые возможности языка C#, для чего и как его применять.
   5. Разработка курсовой работы с целью тестирования паролей.

**2.Алгоритм реализации проекта**

|  |
| --- |
| Выбор среды разработки |
| ↓ |
| Выбор языка программирования |
| ↓ |
| Определение методов перебора |
| ↓ |
| Сравнение методов перебора |
| ↓ |
| Реализация методов перебора |
| ↓ |
| Тестирование |
| ↓ |
| Написание пояснительной записки |

**3. Выбор языка программирования**

C# – объекно-ориентированный язык программирования, действительно интересный инструмент, достойный внимания. Он уверенно занимает высокие позиции в рейтингах востребованных языков программирования на рынке труда. Разработан в 1998 – 2001 годах группой инженеров компании Microsoft и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft/NET Framework. Впоследствии был стандартизирован как ЕСМА-334 и ISO/IEC 23270.

Языки программирования тесно связаны друг с другом таким образом, что на каждый новый язык оказывают в той или иной форме влияние его предшественники. В ходе этого процесса свойства одного языка приспосабливаются к другому языку, полезные нововведения внедряются в существующий контекст, а устаревшие конструкции удаляются. Таким путем развиваются языки программирования и совершенствуется искусство программирования.

**3.1 Сравнение языков программирования**

На старте проекта часто может возникнуть вопрос о выборе средств разработки и если в одних случаях ответ на этот вопрос следует из потребности совместимости с библиотеками и приложениями, то в других ответ не очевиден. Автор берёт во внимание два языка программирования таких как С++ и С#.

С# позволяет стартовать разработку быстрее, а это позволяет быстрее получить прототип решения. Скорость разработки на С# на начальных этапах проекта значительно выше по сравнению с С++.  
Однако, когда инфраструктура проекта создана, основные подходы и библиотеки выбраны, а билд настроен, скорость разработки на С++ и скорость разработки на С# становятся примерно одинаковыми.

Но стоит понимать, что при выборе неправильного подхода, на С++ вполне можно написать код, который будет работать медленнее кода на C#, выполняющего туже задачу.

Стоит сказать, что более сложный код часто легче пишется и анализируется, если написан более простым языком. С этой позиции, используя С#, меньше шансов допустить ошибку в принципиально сложном коде и больше шансов написать чистый код, обладая теми же ресурсами. Это может быть полезно при решении достаточно сложных, но не требовательных к производительности задач.

Таким образом, выбор автора обусловлен относительно одинаковой скоростью работы С# в сравнении с С++ , а также приятным синтаксисом С#.

**4. Выбор среды разработки**

Выбор среды разработки является одной из самых главных аспектов разработки того или иного приложения. Автор отдаёт предпочтение такой среде разработки как Visual Studio , так как она :

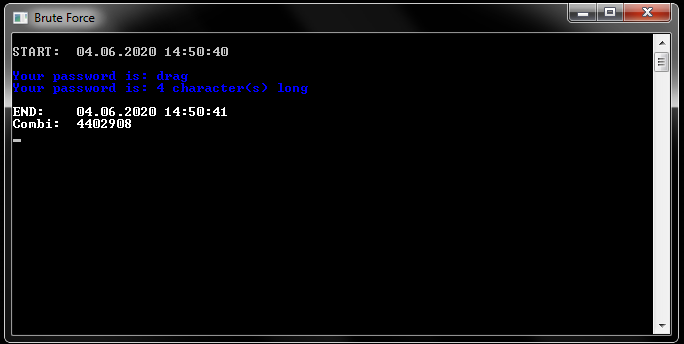
* Официальная. Так как и язык, и среда разработки созданы в Microsoft, логично предположить, что ничего более функционального вы не найдете во всем Интернете. В некоторых случаях без Visual Studio не обойтись — например, при использовании технологий UWP и WPF.
* Бесплатная. Версии «Community edition» для рядового пользователя будет достаточно. Тем более, теперь можно подключать плагины (в отличие от старой версии Express).
* Функциональная. В Visual Studio множество качественных плагинов. С их помощью можно расширить функциональность приложения и подключить другие языки.
* Поддерживает платформы .NET. Visual Studio имеет широкие возможности по разработке приложений под Windows, в том числе в .NET-сегменте.
* Облачные хранилища. Зарегистрируйтесь в сообществе Visual Studio — и получите доступ к облачному хранилищу, где сможете располагать файлы проектов.
* Корпоративность. Технология бэклога позволяет членам команды взаимодействовать при гибкой методологии разработки.

**5.Аспекты программного обеспечения**

**5.1 Функционал программы**

Функционал программы заключается в полном переборе комбинаций всех символов, входящих в масив до тех пор пока комбинация не будет равна паролю, который вы вводите перед тем как запустить основную функцию программы. Также программа позволяет вам узнать за какой промежуток времени был подобран пароль, а также какое колличество комбинаций было совершено перед тем как пароль был подобран.





**5.2 Схема метода перебора**

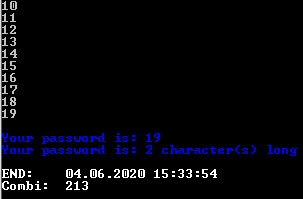
Автор рассматривает работу перебора программы на примере пароля :”19”.  
Сперва функция проходит по одному символу в масиве, который включает в себя английский алфавит, цифры, знаки препинания , а также знаки вычисления   


После чего функция берёт первый символ из масива символов и “скрещивает” его с самим собой , а также со всеми последующими символами масива.



Функция не находит подходящую комбинацию под пароль автора и переходит на второй символ из масива и “скрещивает” его со всеми символами масива. В результате чего, функция находит нужную комбинацию

Видео с процессом подбора комбинации для пароля из данного примера можно найти на сайте <https://github.com/Pavel-Freeman/BruteForce>



**6.Тестирование**

Для проверки правильной работоспособности программы проводяться тесты в виде записи символов, которые отсутствуют в масиве символов, такие как символы кирилицы и другие.

**7.Код игровой программы**

Для написания кода программа была использована рекурсивная функция, с помощью которой процесс подбора правильной комбинации происходил бы быстрее. Код программы предоставлен на сайте <https://github.com/Pavel-Freeman/BruteForce>

**ЗАКЛЮЧЕНИЕ**

Итогом курсовой работы стало

Программное обеспечение для полного перебора комбинаций «BruteForce», полученный с использованием среды разработки Visual Studio.

В процессе проведения курсовой работы изучены возможности применения языка C#.

В результате исследования были выявлены следующие недостатки полученного программного обеспечения:

* + 1. Малая востребованность
    2. Размещение ПО на базе консоли;

В результате программное обеспечение по первому пункту требует больших улучшений, а именно : добавления новых функций, возможность взаимодействовать с другими приложениями, такими как WinRaR , 7zip и др.

Кроме всего перечисленного , продукт нуждается в переносе с консольного в оконный вид , а также разработке дружелюбного интерфейса

Достоинствами данного продукта является следующее:

1.Простота в использовании

2.Возможность проверки надёжности своего пароля в случае попытки атаки BruteForce

Данный игровой продукт может использоваться в определенном круге обращения.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Шилдт, Герберт. C# 4.0: полное руководство.: Пер. с англ. − М.: ООО «И.Д. Вильямс», 2011..

Куликов, С. C. Тестирование программного обеспечения. Базовый курс.: Минск: Четыре четверти, 2017..