**Салон инноваций, изобретение технологий**

Номинация «Информационно - коммуникационные технологии»

**Автоматизированное рабочее место жюри**

**по олимпиадному программированию**

**Выполнил:**

Щемель Дмитрий Артемьевич,

ГБОУДОД СОЦДЮТТ,

МБОУ СМАЛ 10 «Д» класс

Г.Самара, Самарской обл.

**Научный руководитель:**

Забержинский Борислав Эдуардович

**Гор. Самара, 2015**

[ВВЕДЕНИЕ 3](#_Toc414981554)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc414981555)

[1.1 Обзор предметной области 5](#_Toc414981556)

[2. ПРОЕКТИРОВАНИЕ 7](#_Toc414981557)

[2.1 Выбор средств разработки 7](#_Toc414981558)

[2.2 Разработка алгоритма 7](#_Toc414981559)

[2.3 Система оценки 8](#_Toc414981560)

[2.4 Система оценки по добавочным параметрам 9](#_Toc414981561)

[3. РАЗРАБОТКА ИНТЕРФЕЙСА ПРОГРАММЫ 9](#_Toc414981562)

[3.1 Вид интерфейса 9](#_Toc414981563)

[4. ТЕСТИРОВАНИЕ ПРОГРАММЫ 11](#_Toc414981564)

[4.1 Создание базы для тестирования 11](#_Toc414981565)

[ЗАКЛЮЧЕНИЕ 13](#_Toc414981566)

[СПИСОК ЛИТЕРАТУРЫ 14](#_Toc414981567)

### ВВЕДЕНИЕ

В современном мире технологий роль программирования крайне важна. Следовательно, так же существует необходимость проверки знаний в процессе обучения программированию. Одним из методов такой проверки является проведение всевозможных олимпиад в данной области, причем под словом «олимпиада» имеется в виду также обычное дружеское соревнование либо школьный экзамен. А поскольку есть олимпиады – должны быть и методы оценивания результатов этих олимпиад.

Представим обычную жизненную ситуацию, ведь программирование и существует для решения жизненных ситуаций. Допустим, в классе, на уроке информатики необходимо провести самостоятельную работу по программированию. Есть два варианта:

1. Дать ученикам задания, сказать им записывать код программы на листках, потом эти листки собрать и вручную сидеть и проверять. Но при проверки более 20-25 работ велика вероятность того, что скажется человеческий фактор и проверяющий просто-напросто допустит ошибку при проверке.
2. Предоставить ученикам компьютеры для выполнения данной работы, и опять же самому ходить и проверять работу каждого ученика. Данный вариант так же не является оптимальным, т.к. учитель скорее всего всё занятие будет ходить по классу от ученика к ученику(если, конечно, нет возможности удаленного просмотра). А если в этот день несколько самостоятельных?

Как быть при проведении олимпиад по программированию городского уровня и выше? Не вручную же это все проверять.

Именно поэтому и существует необходимость в автоматизации проверки и оценивания результатов олимпиад по программированию. Необходимо создать комплекс программного обеспечения, который смог бы при минимальном человеческом участии собирать ответы программ учеников и сравнивать их с «эталоном». Таким образом, вся работа учителей и жюри будет сводиться к выдаче участникам задания, наблюдению за процессом проведения олимпиады и награждению участников.

**Цель: Создать комплекс программного обеспечения для проведения и оценивания результатов олимпиады по программированию.**

**Задачи:**

1. Изучить уже существующие решения по данному вопросу, а также провести их анализ и сравнение.
2. Разработать методику и алгоритм оценивания результатов проведения олимпиады по программированию.
3. Создать интерфейс необходимой нам программы.

### 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

#### 1.1 Обзор предметной области

Из существующих решений по этой теме, внимания достойно, пожалуй, лишь три сайта.

А именно:

* <http://acm.timus.ru/>
* <http://contest.samara.ru/>
* <http://acmp.ru/>

Рассмотрим каждый из них (таблица 1):

* <http://acm.timus.ru/> - обладает самым большим архивом задач(2500) и имеет широкий выбор компиляторов.
* <http://acmp.ru/> - обладает довольно скромным по сравнению с Тимусом архивом(700 задач) и имеет небольшой выбор компиляторов.
* <http://contest.samara.ru/> - с недавних времен тоже обладает широким выбором компиляторов и часто проводит олимпиады. Однако, имеет серьезный минус – очень часто недоступен.

Таблица 1. – Сравнение существующих решений

|  |  |  |  |
| --- | --- | --- | --- |
| Сайт | Архив задач | Список доступных языков | Дата последней олимпиады |
| <http://acm.timus.ru/> | 2500 | С/С++  Pascal  Java  C#  Go  Python  Ruby  Haskell  Scala | 15 ноября 2014 |
| <http://acmp.ru/> | 700 | Delphi  C++  Java  Visual Basic | 14 февраля 2015 |
| <http://contest.samara.ru/> | 284 | C  С++  С#  Java  Perl  Python  Pascal | 8 февраля 2015 |

Но всё это веб-серверные решения, которые требуют подключения к интернету и не имеют возможности проводить олимпиады на своих условиях. Именно поэтому, целью своей работы я ставлю создания комплекса ПО для проведения и оценивания олимпиад по программированию, но с локальным решением.

### 2. ПРОЕКТИРОВАНИЕ

#### 2.1 Выбор средств разработки

На данный момент существует множество языков программирования и сред разработки. Вот самые известные из них:

* Visual Studio 2013 + C#
* Visual Studio 2013 + C++
* Delphi + Object Pascal
* Eclipse + Java
* Visual Studio 2013 + Visual Basic.Net

Среди предложенных вариантов был выбран первый т.к. IDE Visual Studio обладает широчайшим функционалом, а язык C# при всей своей простоте остается очень мощным и гибким языком.

#### 2.2 Разработка алгоритма

У нас имеется:

1. Набор тестовых заданий
2. Исполняемые файлы, созданные учениками при решении задач.
3. Часть тестовых заданий, предоставленная ученикам.

Как мы видим из представленной схемы (см. рис.1) ученик, использую предоставленное ему условие и примеры и input/output создает запускающий .exe файл. Далее, используя этот .exe файл система тестирует решение для остальных тестовых заданий, поочередно применяя его к каждому тестовому input.txt, и сравнивая содержимое полученного output.txt с верным ответом (для данного input.txt), на основе чего и делается вывод, правильно/неправильно/частично правильно решена данная задача. И так для каждой задачи каждого ученика.

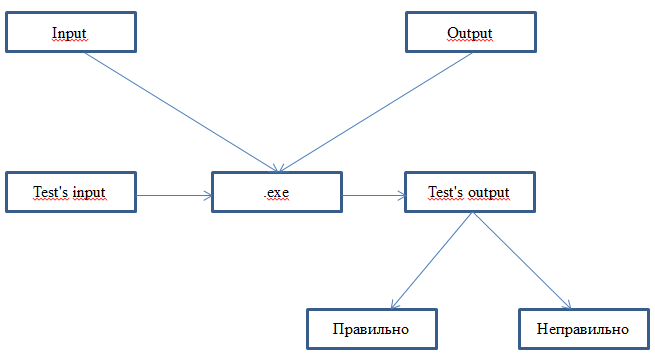


Рисунок 1 - Алгоритм работы программы

#### 2.3 Система оценки

В данной программе будет представлен выбор из двух системы оценки:

1. Система «бал/не бал» – система, в которой ставится 1 бал в случае, если программа прошла успешно все тесты, и ставится 0, если хотя бы один тест был завален.
2. Расширенная система, в которой пользователь данной программы(он же проверяющий) вводит максимальный бал для ученика. Этот максимальный бал делится на количество задач, тем самым мы получаем максимальный балл для данной задачи. Далее за каждый успешной пройденный тест для данной задачи начисляем участнику олимпиады полученный, разделенный на количество тестов для данной задачи.

где n – количество задач, а , где *k* - количество тестов для задачи n;

*MaxBall* *-* Максимальный балл для ученика по всем задачам;

*NumOfTasks* -Количество задач;

*NumOfTests –* Количество тестов для задачи n;

#### 2.4 Система оценки по добавочным параметрам

Дополнительные возможности программы:

* Начисление дополнительных баллов ученику, при правильном решение всех задач, за каждую минуту до окончания олимпиады
* Списывание баллов с ученика за конкретную задачу, если к ней не было предоставлено исходных кодов
* Начисление баллов ученикам, если задание было понято неверно или оно оказалось слишком сложным для них. Начисление происходит только в том случае, если определенная часть учеников дала абсолютно одинаковые ответы. Начисление баллов не происходит ученикам, полностью не справившимся с заданием. Размер компенсации равен разнице между максимальным количеством баллов для задачи и количеством баллов у ученика, полученного за эту же задачу учеником.

### 3. РАЗРАБОТКА ИНТЕРФЕЙСА ПРОГРАММЫ

#### 3.1 Вид интерфейса

**Главное окно программы:**

Интерфейс программы представлен на рисунке 2.1 и 2.2

1. Кнопка, вызывающая окно, в котором задаются параметры работы программы
2. Кнопка, запускающая работу программы
3. Область вывода детальных результатов проверки
4. Область, в которой выводится исходный код решения ученика

5.1 и 5.2) Выбор участника и задачи, для вывода исходных кодов в поле (4)

6) Промежуточная таблица результатов

7) Итоговая таблица результатов

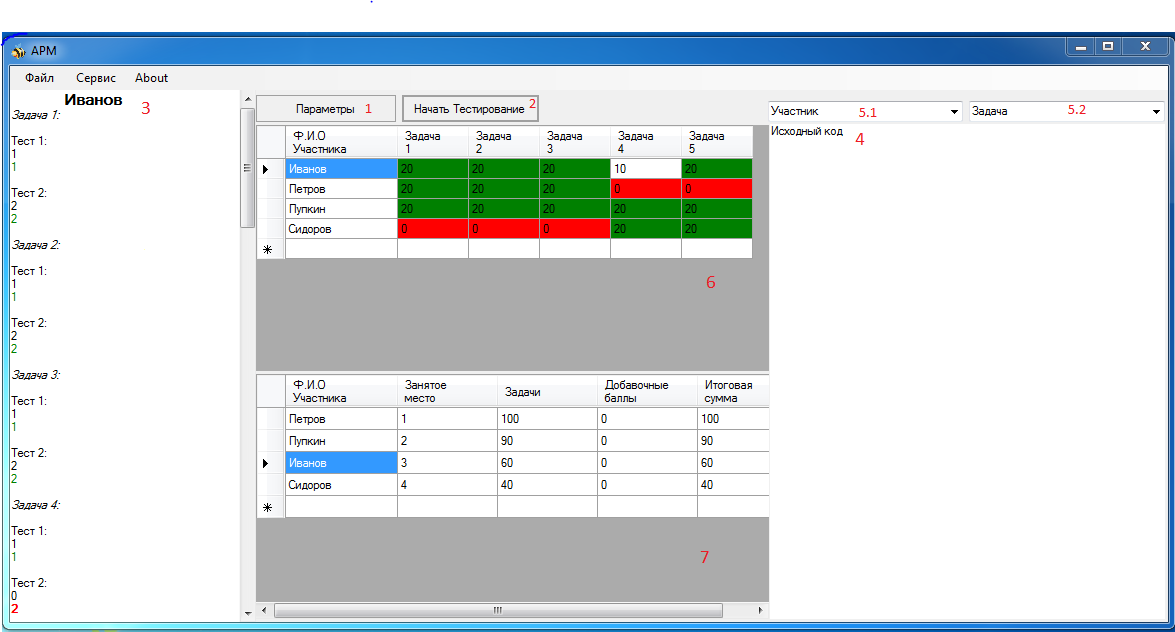


Рисунок 2.1 - Главное окно программы

**Окно параметров работы программы:**

8.1) Кнопка для указания директории с тестовыми наборами

8.2) Кнопка для указания директории с ответами к тестовым наборам

8.3) Кнопка для указания директории с решениями и исходниками учеников

8.4) Кнопка для указания директории, путь которой аналогичен папке с тестовыми наборами, которые были предоставлены ученикам

8.5) Кнопка для указания директории, путь которой аналогичен папке с ответами к тестовым наборам, которые были предоставлены ученикам

9) Поле для указания максимального времени работы программы ученика (едино для всех задач)

10) Область выбор системы оценки

10.1) Поле для указания максимального балла для ученика (для расширенной системы)

11) Область для настройки начисления дополнительных баллов за более быстрое решение

12) Область для настройки списания баллов за отсутствие исходных кодов

13) Область для настройки компенсации баллов за некорректно предоставленные задания

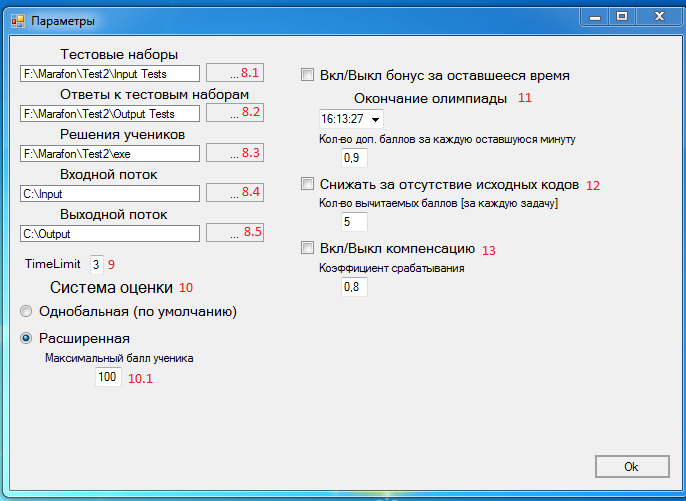
****

Рисунок 2.2 – Окно параметров работы программы

### 4. ТЕСТИРОВАНИЕ ПРОГРАММЫ

#### 4.1 Создание базы для тестирования

В качестве базы для тестирования создадим следующие конфигурации:

I)

* 4 участника со следующими фамилиями: Иванов, Петров, Пупкин и Сидоров.
* 3 задания, содержащие различное количество тестовых наборов.

В первом здание пускай содержится 6 тестовых наборов, во втором – 4 и в третьем тоже 4.

Допустим, что участник Иванов оказался самым умным и безошибочно решил все задания. Участник Петров решил безошибочно два задания, а третье не сделал вообще. Участник Пупкин не справился ни с одной задачей, но решил подшутить над проверяющим и сбросил .exe файл, в котором запускается бесконечный цикл. У участника Сидорова первая программа безошибочно прошла 4/6 тестов, вторая – 3/4, а третья – 2/4.

* Оценка будет происходить по «однобальной» системе, без учета добавочных параметров

Результат работы программы вы можете дать на рисунке 3.

II)

* Те же 4 участника
* 5 заданий, в каждом из которых будет всего по два тестовых набора, но оценивать теперь по «расширенной» системе
* С включенной функцией поощрения за быстрое решение
* С включенной функцией снижения за отсутствия исходных кодов

Допустим, что участник Пупкин безошибочно решил все задания. Участник Петров не решил четвертое и пятое задание (остальные сделал безошибочно). Участник Сидоров сделал только четвертое и пятое (но решил их безошибочно). Участник Иванов сделал все задачи, но четвертое задание прошли только по одному тесту из двух.

Результат работы программы вы можете дать на рисунке 4.

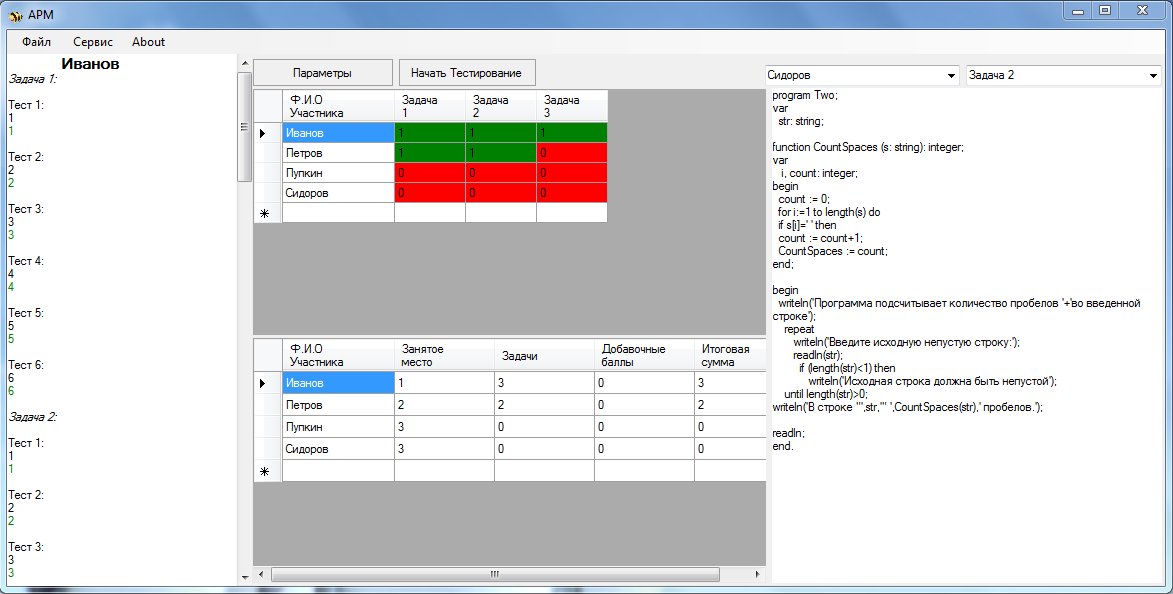


Рисунок 3 - Результат работы программы при первых тестовых условиях

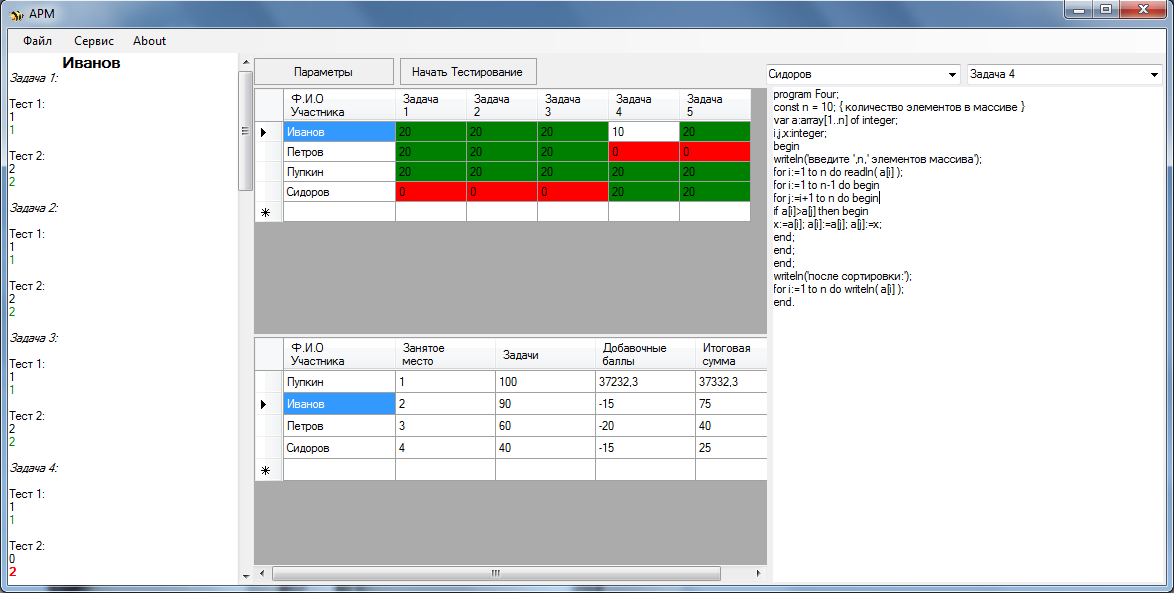


Рисунок 4 - Результат работы программы при вторых тестовых условиях

### ЗАКЛЮЧЕНИЕ

Современное развитие информационных технологий заставляет двигаться современную молодежь в направление изучения информационных технологий и компьютерной грамотности. Высшей степенью освоения теоретического материала является практическая составляющая в виде участия в олимпиадном движении. Из-за возросшего интереса к олимпиадам последнее время наблюдается огромное количество участников на городских, областных и всероссийских конференциях, что, несомненно, создаёт дополнительную нагрузку на членов комиссии (членов жюри). Для достижения постоянного высокого качества результатов оценивания приходится уделять много временных ресурсов на каждую из работ с учетом усталости к последней работе жюри может понизить уровень своей компетенции. Поэтому требуется автоматизированная система, позволяющая на основе методологических подходов жюри производить быструю, точную и независимую оценку. На сегодняшний день такие средства представлены в достаточном количестве в виде коммерческого программного обеспечения(ПО). Но, имея в своём составе алгоритмы и механизмы оценивая и отсутствия исходных кодов, не позволяет производить адаптацию под текущие задачи. Предлагаемый программный комплекс позволит произвести, в условиях локальной сети, олимпиаду по программированию и оценить результаты. Несмотря на ограниченный функционал у системы имеется перспектива дальнейшей модернизации, в целях повышения точности оценки и дополнительных параметров.

### СПИСОК ЛИТЕРАТУРЫ

1. <http://msdn.microsoft.com/>
2. Агуров, Павел. C#. Разработка компонентов вMS Visual Studio 2013 ( БХВ-Петербург, 2014 )
3. Культин. Microsoft Visual C# в задачах и примерах ( БХВ-Петербург, 2014 )
4. СП гос университет инф технологий. Всероссийская командная олимпиада школьников по информатике и программированию (ред, Васильев, 2011 )
5. Гу ИТМО,. Десятая Всероссийская олимпиада школьников по информатике и программированию. ( Гу ИТМО, 2009 )
6. Шилдт. C# учебный курс (БХВ-Петербург, 2015 )
7. C++ и Visual Studio. NET. Самоучитель программиста Баженова ( КУДИЦ-ОБРАЗ, 2010
8. Васильев. C#. Объектно-ориентированное программирование. Учебный курс (Питер-Юг, 2012 )
9. МГТУ им Баумана. Объектно - ориентированное программирование. Иванова, Ничушкина, Пугачев ( МГТУ им Баумана, 2009 )
10. СГОС.Объектно-ориентированное программирование (СГОС, 2012 )