# ПРАКТИЧЕСКАЯ РАБОТА № 1.0

**Тема:** Введение. Основы работы с Git.

**Цель:** Научиться использовать систему контроля версий при разработке приложений.

**Время выполнения:** 6 часов.

## 1 Входной контроль

1. Класс – это элемент ПО, описывающий абстрактный тип данных, а также его частичную или полную реализацию.
2. Конструкторы – функция, предназначенная для инициализации объектов класса. Деструктор – функция, предназначенная для деинициализации объекта.
3. Модификаторы доступа позволяют задать допустимую область видимости для членов класса. Применяются следующие модификаторы доступа:

* Public. Публичный, общедоступный класс или член класса. Такой член класса доступен из любого места в коде, а также из других программ и сборок.
* Private. Закрытый класс или член класса. Представляет полную противоположность модификатору public. Такой закрытый класс или член класса доступен только из кода в том же классе или контексте.
* Protected. Такой член класса доступен из любого места в текущем классе или в производных классах. При этом производные классы могут располагаться в других сборках.

## 2 Ход выполнения работы

## 2.1 Задание № 1. Установка Git и регистрация на сайте github.com/

Если система контроля версий Git не установлена, то установить ее (параметры оставить по умолчанию). При отсутствии аккаунта на сайте https://github.com/ произвести регистрацию (рисунок 2.1.1).

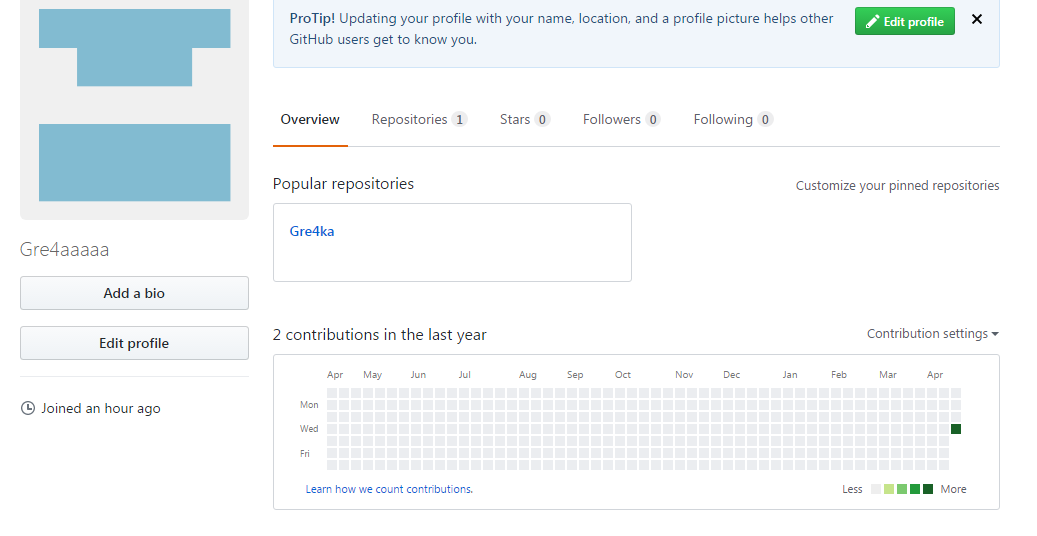


Рисунок 2.1.1 – Зарегистрированный аккаунт на сайте github.com/

## 2.2 Задание № 2. Создание проекта в Visual Studio для реализации индивидуального задания

Создание проекта в Visual Studio и индивидуальное задание отображены на рисунках 2.2.1 – 2.2.2. Реализация варианта показана на рисунках 2.2.3 – 2.2.6.

## 

Рисунок 2.2.1 – Создание проекта в Visual Studio

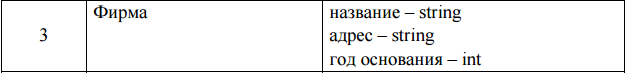


Рисунок 2.2.2 – Вариант индивидуального задания

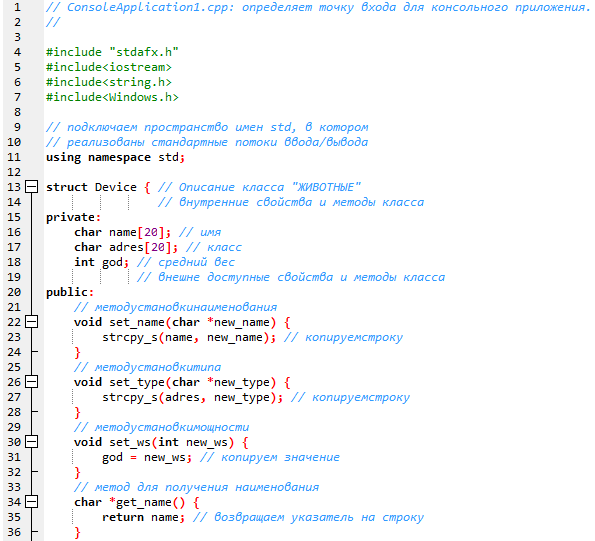


Рисунок 2.2.3 – Код программы индивидуального задания



Рисунок 2.2.4 – Код программы индивидуального задания

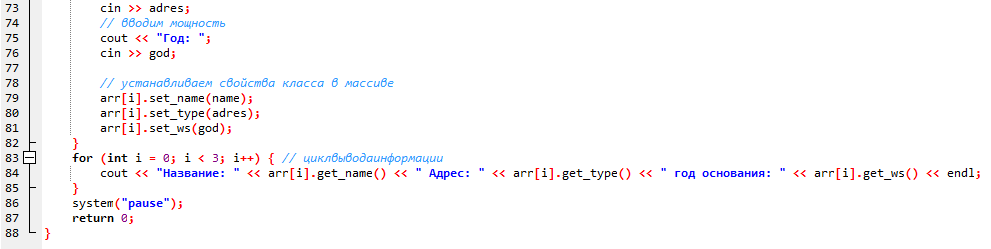


Рисунок 2.2.5 – Код программы индивидуального задания

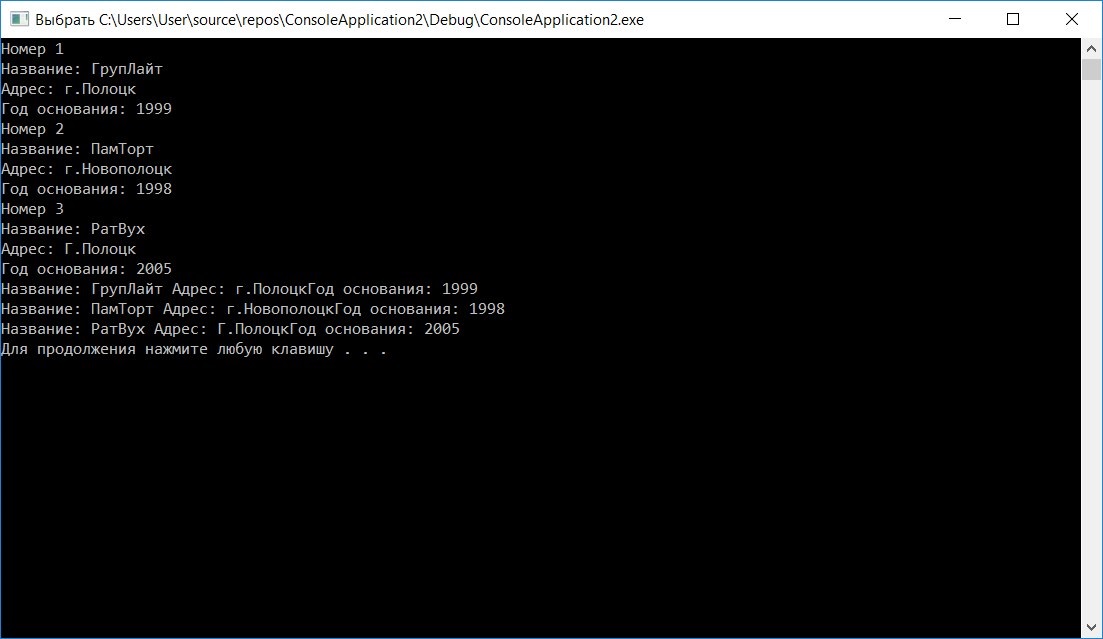


Рисунок 2.2.6 – Проверка работоспособности программы

## 2.3 Задание № 3. Запуск Git Bash, создание новой локальной репозитории. Фиксирование состояния репозитория

Создание новой локальной репозитории и фиксирование состояния репозитория отображены на рисунках 2.3.1–2.3.2.

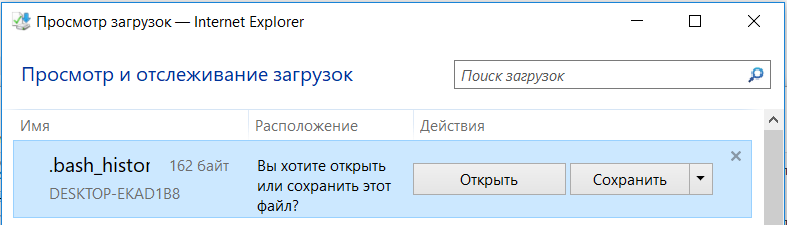


Рисунок 2.3.1 – Созданные загрузки GitBash

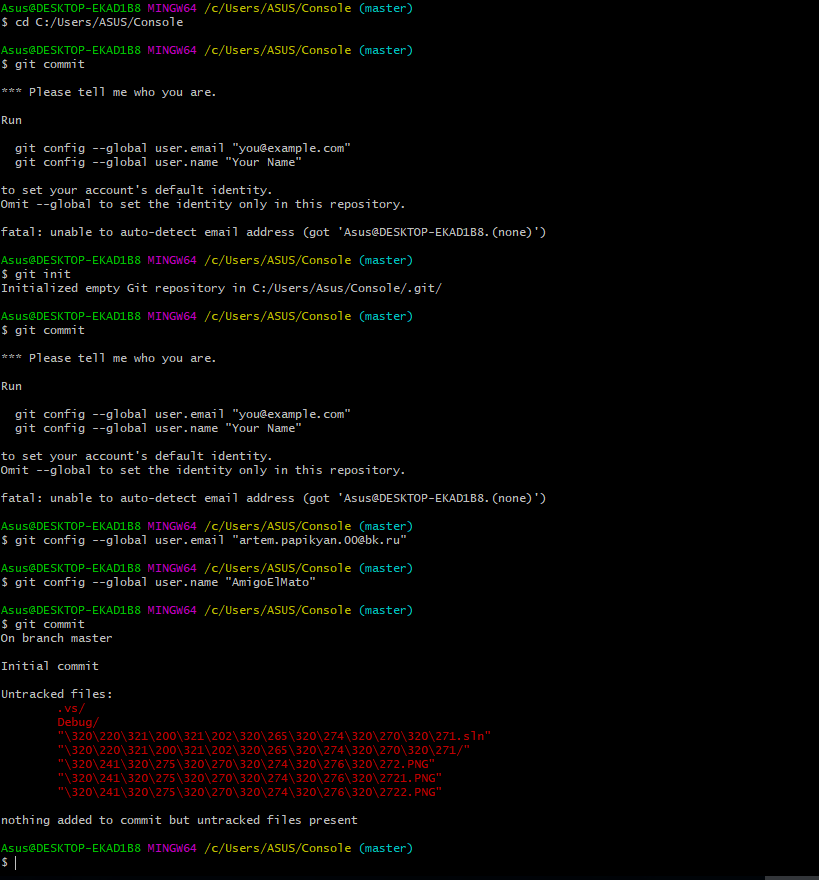


Рисунок 2.3.2 – Создание новой локальной репозитории

## Вывод: в ходе практической работы мною были усовершенствованы знания, умения и навыки при работе с программой GitBash, а также использование системы контроля версий при разработке приложений. Также, был реализован класс согласно варианту задания. Цель и задачи работы были достигнуты.