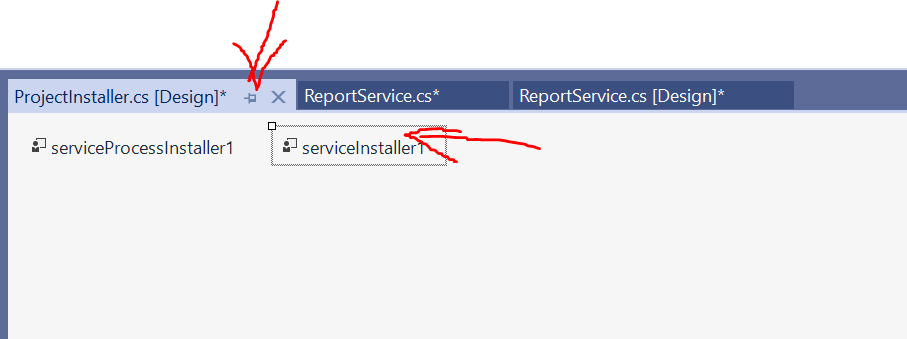
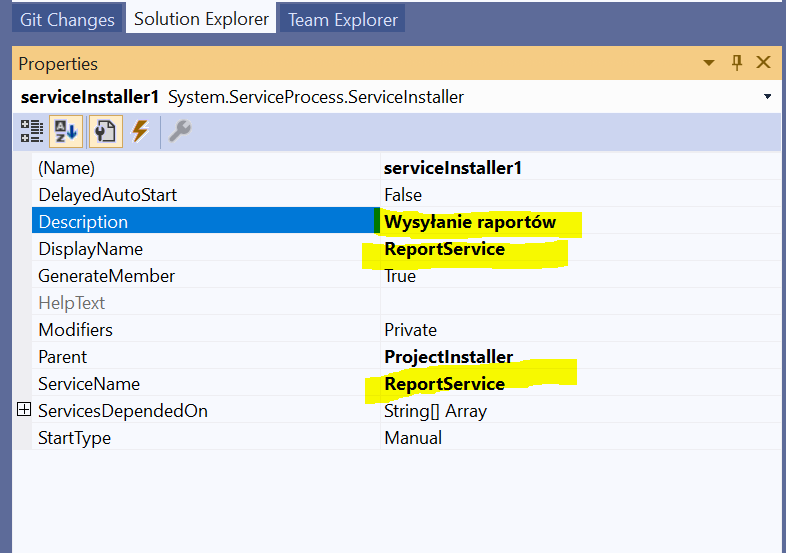
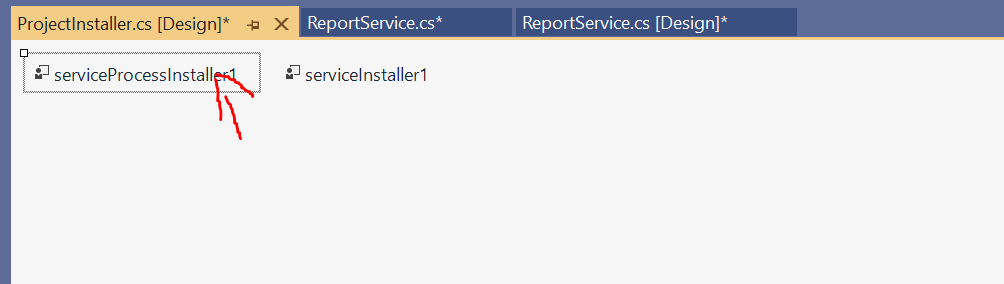
Aplikacja, która nie posiada interfejsu użytkownika, nie podejmuje żadnej interakcji z użytkownikiem.

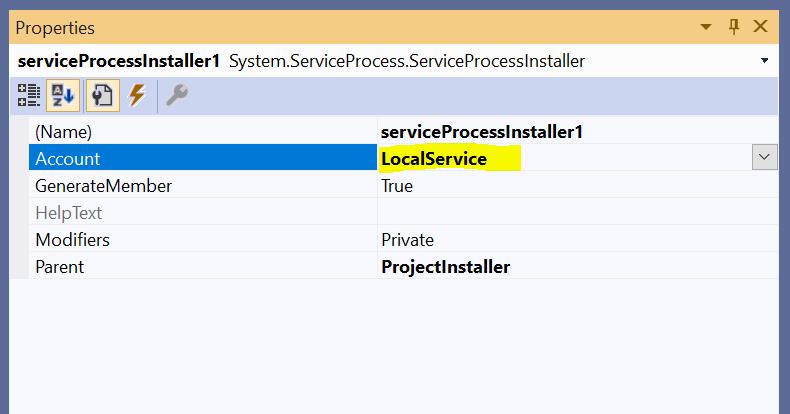
Service1.cs F7 lub View Code  
OnStart()  
OnStop()

Na oknie Designer’a wybieramy z menu Add Installer









Modyfikujemy metodę Main w klasie program

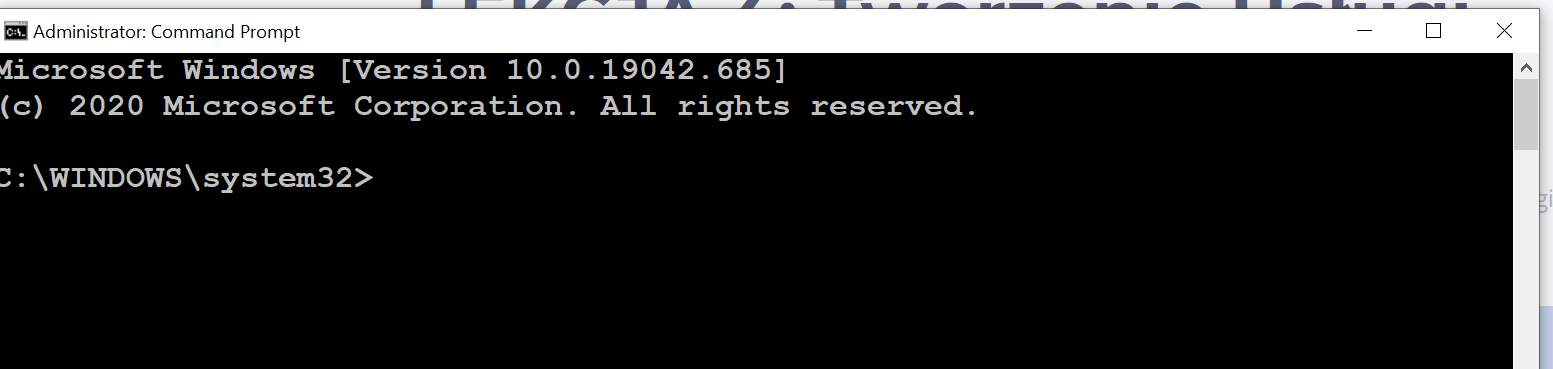
|  |
| --- |
| static void Main(string[] args)  {  ServiceBase[] ServicesToRun;  ServicesToRun = new ServiceBase[]  {  new ReportService()  };  ServiceBase.Run(ServicesToRun);  } |

|  |
| --- |
| ServiceBase[] ServicesToRun;  ServicesToRun = new ServiceBase[]  {  new ReportService()  };  ServiceBase.Run(ServicesToRun); |

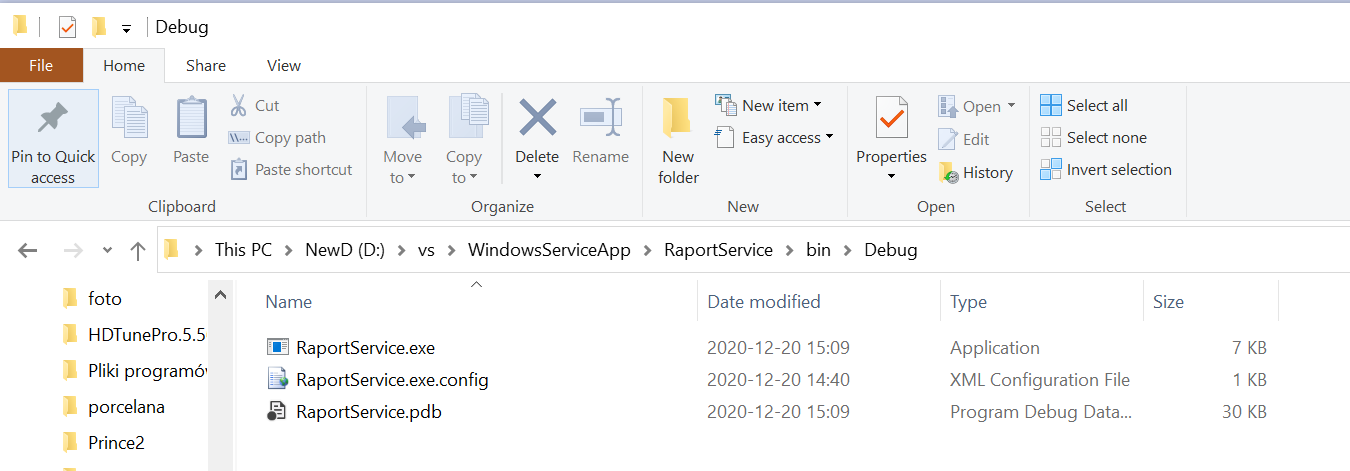
|  |
| --- |
| if (Environment.UserInteractive)  {  var parameter = string.Concat(args);  switch (parameter)  {  case "--install":  ManagedInstallerClass.InstallHelper(new[]  {Assembly.GetExecutingAssembly().Location});  break;  case "--uninstall":  ManagedInstallerClass.InstallHelper(new[]  {"/u",Assembly.GetExecutingAssembly().Location});  break;  default:  break;  }  }  else  {  ServiceBase[] ServicesToRun;  ServicesToRun = new ServiceBase[]  {  new ReportService()  };  ServiceBase.Run(ServicesToRun);  } |

Kompilujemy program

Uruchamiamy cmd z uprawnieniami administratora

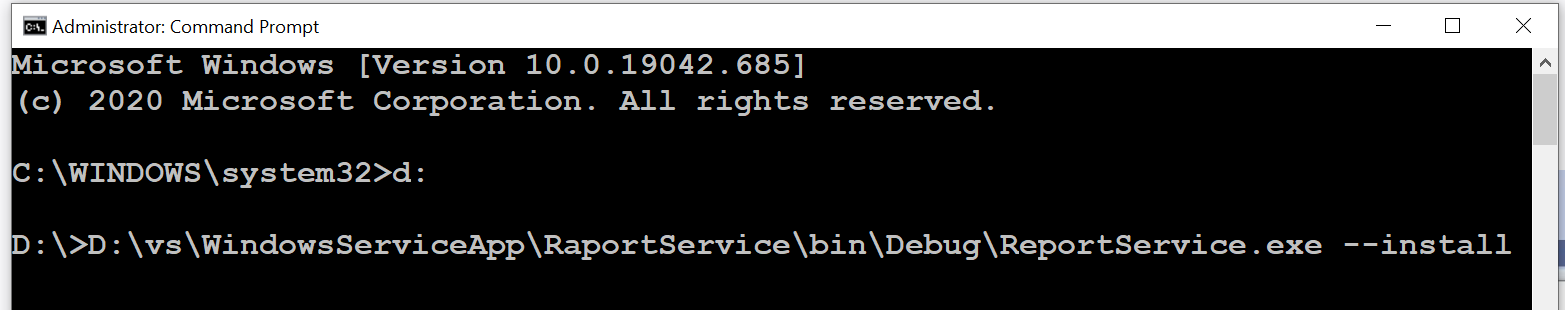


Na projekcie open in file explorer,  
przechodzimy do bin, następnie do debug



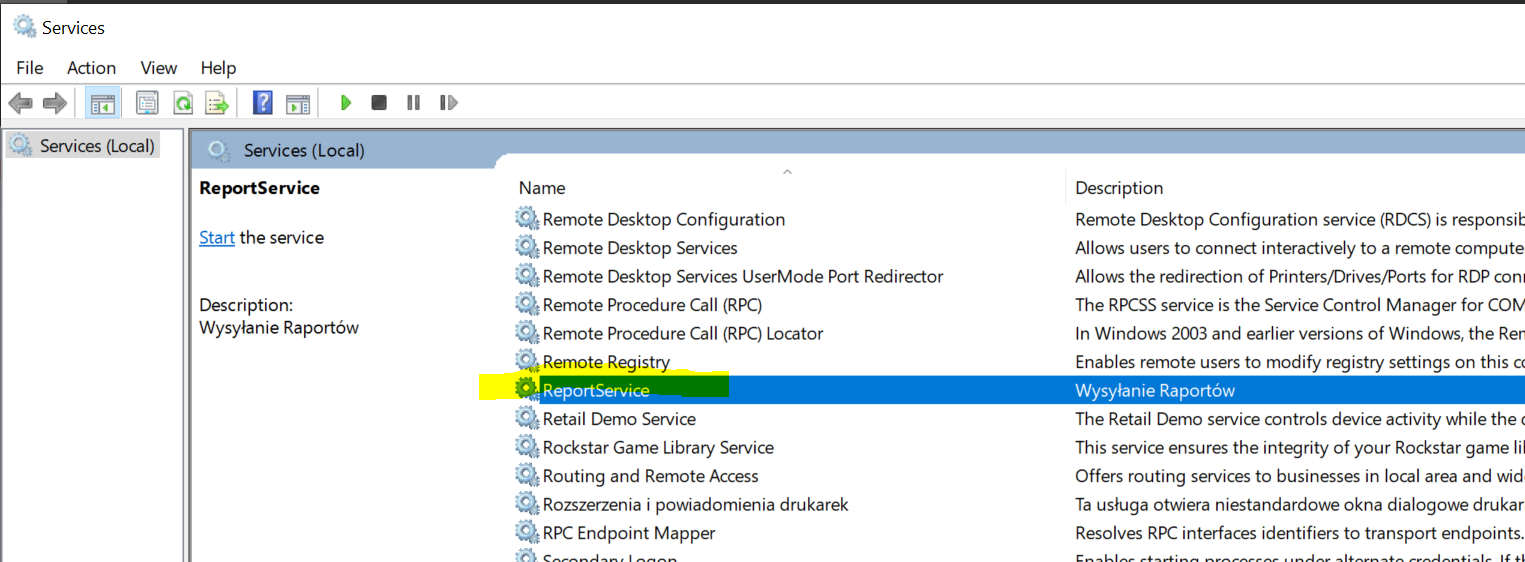
Wklejamy ścieżkę, dopisujemy nazwę naszego exeka i dopisujemy –install

D:\vs\WindowsServiceApp\ReportService\bin\Debug\ReportService.exe --install



Powstaje plik : ReportService.InstallLog

Uruchamiamy services.msc



D:\vs\WindowsServiceApp\ReportService\bin\Debug\ReportService.exe –uninstall

**Implementacja Logiki Biznesowej**

1 dodajemy klasę timer