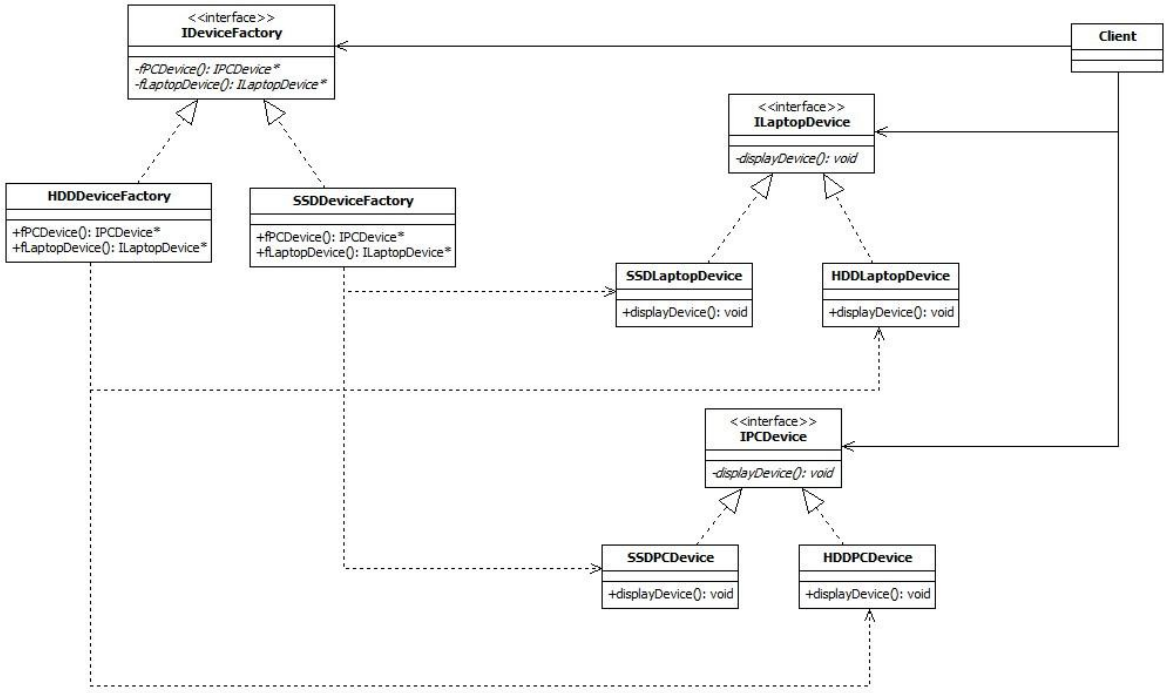
# LB 14 - “Fabryka Abstrakcyjna”

Krystian Kogut, nr albumu 308128

## Diagram klas




## Kod

```
#if !defined(_CLIENT_H)
#define _CLIENT_H

class Client {
};

#endif  //_CLIENT_H
```

```h
#if !defined(_HDDDEVICEFACTORY_H)
#define _HDDDEVICEFACTORY_H

#include "IDeviceFactory.h"
#include "IPCDevice.h"
#include "ILaptopDevice.h"
#include "IPCDevice.h"

class HDDDeviceFactory : public IDeviceFactory {
public:
    IPCDevice* fPCDevice();
    ILaptopDevice* fLaptopDevice();
};

#endif  //_HDDDEVICEFACTORY_H
```

```cpp
#include "IPCDevice.h"
#include "HDDPCDevice.h"
#include "HDDLaptopDevice.h"
#include "ILaptopDevice.h"
#include "HDDDeviceFactory.h"

IPCDevice* HDDDeviceFactory::fPCDevice() {
    return new HDDPCDevice();
}

ILaptopDevice* HDDDeviceFactory::fLaptopDevice() {
    return new HDDLaptopDevice();
}
```

```h
#if !defined(_HDDLAPTOPDEVICE_H)
#define _HDDLAPTOPDEVICE_H

#include "ILaptopDevice.h"


class HDDLaptopDevice : public ILaptopDevice {
public:
    void displayDevice();
};

#endif  //_HDDLAPTOPDEVICE_H
```

```cpp
#include "HDDLaptopDevice.h"

void HDDLaptopDevice::displayDevice() {
    std::cout<<"HDD Laptop Device" <<std::endl;
}
```

```cpp
#if !defined(_HDDPCDEVICE_H)
#define _HDDPCDEVICE_H

#include "IPCDevice.h"

class HDDPCDevice : public IPCDevice {
public:
    void displayDevice();
};

#endif  //_HDDPCDEVICE_H
```

```cpp
#include "HDDPCDevice.h"

void HDDPCDevice::displayDevice() {
    std::cout<<"HDD PC Device" <<std::endl;
}
```

```cpp
#if !defined(_IDEVICEFACTORY_H)
#define _IDEVICEFACTORY_H

#include "IPCDevice.h"
#include "ILaptopDevice.h"
#include <iostream>

class IDeviceFactory {
public:
    virtual IPCDevice* fPCDevice() = 0;
    virtual ILaptopDevice* fLaptopDevice() = 0;
};

#endif  //_IDEVICEFACTORY_H
```

```cpp
#if !defined(_ILAPTOPDEVICE_H)
#define _ILAPTOPDEVICE_H
#include <iostream>

class ILaptopDevice {
public:
    virtual void displayDevice() = 0;
};

#endif  //_ILAPTOPDEVICE_H
```

```cpp
#if !defined(_IPCDEVICE_H)
#define _IPCDEVICE_H
#include <iostream>

class IPCDevice {
public:
    virtual void displayDevice() = 0;
};

#endif  //_IPCDEVICE_H
```

```cpp
#if !defined(_SSDDEVICEFACTORY_H)
#define _SSDDEVICEFACTORY_H

#include "IDeviceFactory.h"
#include "IPCDevice.h"
#include "ILaptopDevice.h"

class SSDDeviceFactory : public IDeviceFactory {
public:
    IPCDevice* fPCDevice();
    ILaptopDevice* fLaptopDevice();
};

#endif  //_SSDDEVICEFACTORY_H
```

```cpp
#include "IPCDevice.h"
#include "SSDPCDevice.h"
#include "SSDLaptopDevice.h"
#include "ILaptopDevice.h"
#include "SSDDeviceFactory.h"


IPCDevice* SSDDeviceFactory::fPCDevice() {
    return new SSDPCDevice();
}

ILaptopDevice* SSDDeviceFactory::fLaptopDevice() {
    return new SSDLaptopDevice();
}
```

```cpp
#if !defined(_SSDLAPTOPDEVICE_H)
#define _SSDLAPTOPDEVICE_H

#include "ILaptopDevice.h"

class SSDLaptopDevice : public ILaptopDevice {
public:
    void displayDevice();
};

#endif  //_SSDLAPTOPDEVICE_H
```

```cpp
#include "SSDLaptopDevice.h"

void SSDLaptopDevice::displayDevice() {
    std::cout<<"SSD Laptop Device" <<std::endl;
}
```

```h
#if !defined(_SSDPCDEVICE_H)
#define _SSDPCDEVICE_H

#include "IPCDevice.h"

class SSDPCDevice : public IPCDevice {
public:
    void displayDevice();
};

#endif  //_SSDPCDEVICE_H
```

```cpp
#include "SSDPCDevice.h"

void SSDPCDevice::displayDevice() {
    std::cout<<"SSD PC Device"<<std::endl;
}
```

```cpp
#include <iostream>
#include "IDeviceFactory.h"
#include "HDDDeviceFactory.h"

int main() {
    IDeviceFactory *iDeviceFactory;
    ILaptopDevice *iLaptopDevice;
    iDeviceFactory = new HDDDeviceFactory;
    iLaptopDevice = iDeviceFactory->fLaptopDevice();
    iLaptopDevice->displayDevice();
    std::cin.get();

    return 0;
}
```

Wyniki programu:

```
HDD Laptop Device


Process finished with exit code 0
```