Budowniczy *(ang. builder)* - wzorzec projektowy, który dostarcza mechanizm do konwersji danych z konkretnego formatu wejściowego do dowolnego formatu wyjściowego. Wzorzec zapewnia łatwy mechanizm dostarczania nowych formatów danych wyjściowych bez konieczności modyfikacji obiektu zarządzającego (tzw. kierownika).

```
class ZestawKomputerowy {
private:
    string monitor;
    string procesor;
    string grafika;
    string ram;
    string hdd;
public:
    void setMonitor(string m) {
        monitor = m;
    }

    void setProcesor(string p) {
        procesor = p;
    }

    void setGrafika(string g) {
        grafika = g;
    }

    void setRam(string r) {
        ram = r;
    }

    void setHdd(string h) {
        hdd = h;
    }

    void show() {
        if (monitor != "") cout << "Monitor = " << monitor << endl;
        if (procesor != "") cout << "Procesor = " << procesor << endl;
        if (grafika != "") cout << "Grafika = " << grafika << endl;
        if (ram != "") cout << "RAM = " << ram << endl;
        if (hdd != "") cout << "HDD = " << hdd << endl;
    }
};
```

```
class Builder {
protected:
    ZestawKomputerowy* zestawKomputerowy;

public:
    void newZestaw() {
        zestawKomputerowy = new
ZestawKomputerowy();
    }

    ZestawKomputerowy getZestaw() {
        return*zestawKomputerowy;
    }

    virtual void buildMonitor() = 0;
    virtual void buildProcesor() = 0;
    virtual void buildGrafika() = 0;
    virtual void buildRam() = 0;
    virtual void buildHdd() = 0;
};
```

```
class ZestawABC996 : public Builder {
public:

    ZestawABC996() :Builder() {
    }

    void buildMonitor() {
        zestawKomputerowy->setMonitor("LG");
    }

    void buildProcesor() {
        zestawKomputerowy->setProcesor("INTEL");
    }

    void buildGrafika() {
        //zestaw nie obejmuje karty graficznej
    }

    void buildRam() {
        zestawKomputerowy->setRam("DDR");
    }

    void buildHdd() {
        zestawKomputerowy->setHdd("Samsung");
    }
};
```

```
class Director {
private:
    Builder* builder;

public:
    void setBuilder(Builder* b) {
        builder = b;
    }

    ZestawKomputerowy getZestaw() {
        return builder->getZestaw();
    }

    void skladaj() {
        builder->newZestaw();
        builder->buildMonitor();
        builder->buildProcesor();
        builder->buildHdd();
        builder->buildRam();
        builder->buildGrafika();
    }
};



int main() {
    Director* szef = new  Director();
    Builder* builder = new  ZestawABC996();

    szef->setBuilder(builder);
    szef->skladaj();
    ZestawKomputerowy zestaw = szef->getZestaw();

    zestaw.show();

    return 0;
}
```