

```

1  #include "bib.h"
2
3  int main()
4  {
5      Powiesc d(7, "Monia Kociolek", "Mordo", "Tim x");
6      d.wyswietl();
7
8      Historyczna h(2137, "Jan Pawel", "tak jak powiedzial", 2003);
9
10     h.wyswietl();
11
12     Naukowa n(54, "Roman", "Lombi", "majster");
13
14     Czytelnik Jakub(69, "kubus", "puchatek");
15
16     Jakub.wypożycz(&d);
17     Jakub.wypożycz(&h);
18     Jakub.wypożycz(&n);
19
20     Jakub.usunNaukowe();
21
22     cout<<Jakub<<endl;
23
24
25
26
27     return 0;
28 }
29
30 #ifndef BIB
31 #define BIB
32
33 #include <iostream>
34 #include <vector>
35 #include <string>
36
37 using namespace std;
38
39 class Ksiazka
40 {
41 private:
42     int id;
43     string autor;
44     string tytul;
45 public:
46     Ksiazka(int id, string autor, string tytul) : id(id),
47     autor(autor), tytul(tytul) {}
48     virtual ~Ksiazka() = default;
49     virtual void wyswietl() const = 0;
50     int getId() const {return id;}

```

```

50     string getautor() const {return autor;}
51     string gettytul() const {return tytul;}
52 };
53
54 class Powiesc : public Ksiazka
55 {
56 private:
57     string wydawca;
58 public:
59     Powiesc(int id, string autor, string tytul, string
wydawca) : Ksiazka(id,autor,tytul), wydawca(wydawca) {}
60     ~Powiesc() {cout<<"\n ~Powiesc";}
61     void wyswietl() const override
62     {
63         cout<<"id: "<<getid()<<"\nAutor: "<<getautor()<<"
Tytul: "<<gettytul()<<"\nwydawca: "<<wydawca<<endl;
64     }
65 };
66
67 class Historyczna : public Ksiazka
68 {
69 private:
70     int rok;
71 public:
72     Historyczna(int id, string autor, string tytul, int rok)
: Ksiazka(id,autor,tytul), rok(rok) {}
73     ~Historyczna() {cout<<"\n ~Historyczna";}
74     void wyswietl() const override
75     {
76         cout<<"id: "<<getid()<<"\nAutor: "<<getautor()<<"
Tytul: "<<gettytul()<<"\nrok: "<<rok<<endl;
77     }
78 };
79
80 class Naukowa : public Ksiazka
81 {
82 private:
83     string dziedzina;
84 public:
85     Naukowa(int id, string autor, string tytul, string
dziedzina) : Ksiazka(id, autor, tytul), dziedzina(dziedzina) {}
86     ~Naukowa() {cout<<"\n ~Naukowa";}
87     void wyswietl() const override
88     {
89         cout<<"id: "<<getid()<<"\nAutor: "<<getautor()<<"
Tytul: "<<gettytul()<<"\ndziedzina: "<<dziedzina<<endl;
90     }
91 };
92
93

```

```

94  /// czytelnik
95
96  class Czytelnik
97  {
98  private:
99      int nr;
100     string imie;
101     string nazwisko;
102     vector<Ksiazka*>lista;
103 public:
104     Czytelnik(int nr,string imie, string nazwisko) :nr(nr),
105     imie(imie), nazwisko(nazwisko) {}
106     ~Czytelnik() = default;
107
108     void wypożycz(Ksiazka* ksiazka)
109     {
110         lista.push_back(ksiazka);
111         cout<<"\nKsiazka wypożyczona przez czytelnika o
112         numerze: "<<nr<<endl;
113     }
114
115     friend ostream& operator<<(ostream& os, const Czytelnik&
116     czytelnik)
117     {
118         os<<"\nCzytelnik o numerze:
119         "<<czytelnik.nr<<"\nImie: "<<czytelnik.imie<<"\nNazwisko:
120         "<<czytelnik.nazwisko<<endl;
121         os<<"\n\nWypożyczone ksiazki:"<<endl;
122         for(const auto* ksiazka : czytelnik.lista)
123         {
124             ksiazka->wyswietl();
125         }
126         return os;
127     }
128
129     void operator--=(int indeks)
130     {
131         lista.erase(lista.begin()+indeks);
132     }
133
134     void usunNaukowe()
135     {
136         for (auto it = lista.begin(); it != lista.end(); )
137         {
138             if (dynamic_cast<Naukowa*>(*it) != nullptr)
139             {
140                 it = lista.erase(it);
141             }
142             else

```

```
139         {
140             ++it;
141         }
142     }
143 }
144
145 };
146 #endif // BIB
147
```