```cpp
/* ------------------- */
/*   Aluno    78865    */
/*   Miguel   Amaral   */
/* Quarta Feira 15h30  */
/* ------------------- */


/* ------------------- */
/*  PaymentMethod.cpp  */
/* ------------------- */
#include <string>

class PaymentMethod {
private:
    std::string  name;
public:
    PaymentMethod(std::string name) :  name(name) {}
    std::string getName() { return  name; }
    void setName(std::string name) {  name = name; }
};

/* ------------------- */
/*    ProductType.cpp  */
/* ------------------- */
#include <string>

class ProductType {
private:
    std::string  name;
public:
    ProductType(std::string name) :  name(name) {}
    std::string getName() { return  name; }
    void setName(std::string name) {  name = name; }
};

/* ------------------- */
/*    Purchase.cpp     */
/* ------------------- */
#include <forward list>
#include <ctime>
#include "ProductType.cpp"
#include "PaymentMethod.cpp"

class Purchase {
private:
    std::time t  date;
    int  value =0;
    std::shared ptr<PaymentMethod>  paymentMethod;
    std::shared ptr<ProductType>  prodType;
public:
    Purchase(std::time t date, std::string payment, int value,
        std::string prodType ) :  date(date),  value(value){
                    paymentMethod = std::make shared<PaymentMethod>(prodType);
                    prodType = std::make shared<ProductType>(prodType);
    }
    int getValue(){
        return  value;
    }
    std::time t getDate(){
        return  date;
    }
    void applyDiscount(float discount){
         value =  value * (1-discount);
    }

};

```

```
/* ------------------- */
/*      Filter.hpp     */
/* ------------------- */
class Filter {
public:
    virtual std::list< std::shared ptr<Client>>
                    getFilteredList(Store *store) =0;
};

/* ------------------- */
/*      Filter.cpp     */
/* ------------------- */
#include <forward list>
#include <list>
//#include "Client.cpp"

class XPurchases : public Filter {
private:
  int  purchaseNumber;
public:
  XPurchases(int purchaseNumber){
     purchaseNumber = purchaseNumber;
  }


  std::list< std::shared ptr<Client>> getFilteredList(Store *store){

    std::cout << "A filtrar PURCHASES " << std::endl;
    std::list< std::shared ptr<Client>> returnList;
    for (int i = 0; i < store->getSize(); i++) {
        std::shared ptr<Client> c = store->getClient(i);
        if(c->getPurchaseListSize() >=  purchaseNumber)
          returnList.push back(std::move(c));
    }
    return returnList;
  }

};

class XPoints : public Filter {
private:
  int  pointsMin;
public:
  XPoints(int pointsMin){
     pointsMin = pointsMin;
  }


  std::list< std::shared ptr<Client>> getFilteredList(Store *store){
    std::cout << "A filtrar POINTS " << std::endl;
    std::list< std::shared ptr<Client>> returnList;
    for (int i = 0; i < store->getSize(); i++) {
        std::shared ptr<Client> c = store->getClient(i);
        if(c->getPoints() >=  pointsMin)
          returnList.push back(std::move(c));
    }
    return returnList;
  }

};

/* ------------------- */
/*     Client.cpp      */
/* ------------------- */
#include <forward list>
#include <list>
#include <ctime>
```

```cpp
#include <string>
#include "Purchase.cpp"

class Client {
private:
    std::string  name;
    long int  contact;
    std::time t  birthday;
    std::time t  firstBuy;
    std::list< std::shared ptr<Purchase> >  purchases;
    int  points;
    float  discount = 0;
public:
    Client(std::time t birthday, long int contact, std::string name) :
                        birthday(birthday),  contact(contact),  name(name){
         points = 0;
    }
    void addPurchase(std::shared ptr<Purchase> p){
         points =  points + p->getValue();
        if( firstBuy){
             firstBuy = p->getDate();
        }
         purchases.push back(std::move(p));
    }

    void setDiscount(double discount){
         discount = discount;
    }

    int getPurchaseListSize(){
        return  purchases.size();
    }

    int getPoints(){return  points;}
    void setPoints(int points){  points = points;}

    std::string getName(){return  name;}


};

/* ------------------- */
/*      Store.cpp      */
/* ------------------- */
#include <string>
#include "Client.cpp"
#include "Filter.hpp"

class Store {
private:
    std::string  name;
    std::list< std::shared ptr<Client> >  clients;
public:
    Store(std::string name) :  name(name){}
    void addClient(std::shared ptr<Client> c){
         clients.push back(std::move(c));
    }

    std::list< std::shared ptr<Client> > getList(std::shared ptr<Filter> filter){
        return filter->getFilteredList(this);
    }

    std::shared ptr<Client> getClient(int index) {
        std::list<std::shared ptr<Client>>::iterator i =  clients.begin();
        std::advance(i, index);
        return *i;
}
```

```
202
203 int getSize() {
204         return  clients.size();
205 }
206 };
207
208 /* ------------------- */
209 /*       App.cpp       */
210 /* ------------------- */
211 #include <forward list>
212 #include <iostream>
213
214 class Store;
215
216 #include "Store.cpp"
217 #include "Filter.cpp"
218
219 int main(){
220   std::shared ptr<Store> store = std::make shared<Store>("loja de ferragens");
221   time t currTime;
222   time t tenYears;
223   time(&currTime);
224   tenYears = currTime - (10*365*24*60*60);
225
226
227   std::shared ptr<Client> c1=std::make shared<Client>(currTime,1111111,"Joao");
228   std::shared ptr<Client> c2=std::make shared<Client>(tenYears,9999999,"Jota");
229   store->addClient(c1);
230   store->addClient(c2);
231
232   std::shared ptr<Purchase> p1 =
233                         std::make shared<Purchase>(currTime, "cash", 30, "milk");
234   std::shared ptr<Purchase> p2 =
235                         std::make shared<Purchase>(currTime, "cash", 30, "eggs");
236   std::shared ptr<Purchase> p3 =
237                         std::make shared<Purchase>(currTime, "cash", 30, "milk");
238   std::shared ptr<Purchase> p4 =
239             std::make shared<Purchase>(currTime, "credit card", 1000, "laptop");
240
241
242   c1->addPurchase(p1);
243   c1->addPurchase(p2);
244   c1->addPurchase(p3);
245
246   c2->addPurchase(p4);
247
248
249   std::shared ptr<Filter> f2 = std::make shared<XPurchases>(2);
250   std::list< std::shared ptr<Client> > listaCompras2 = store->getList(f2);
251
252   std::list<std::shared ptr<Client>>::iterator i;
253   for (i = listaCompras2.begin(); i != listaCompras2.end(); ++i) {
254       std::shared ptr<Client> cliente = *i;
255       std::cout << cliente->getName() << std::endl;
256
257   }
258
259   std::shared ptr<Filter> f3 = std::make shared<XPoints>(100);
260   std::list< std::shared ptr<Client> > listaPontos400 = store->getList(f3);
261   for (i = listaPontos400.begin(); i != listaPontos400.end(); ++i) {
262       std::shared ptr<Client> cliente = *i;
263       cliente->setDiscount(0.50);
264       std::cout << cliente->getName() << std::endl;
265   }
266   return 0;
267 }
268
```