código

1.

```
int main() {
  string buffer;
  junta(buffer, "Bom dia!"); //buffer passa a ter "Bom dia!" no seu interior
  junta(buffer, "ola"); //buffer passa a ter "Bom dia! ola"
  junta(buffer, " outra ", "vez!"); //buffer passa a "Bom dia! ola outra vez!"
  junta(buffer, "...") = "novo conteúdo"; //no fim da execução desta linha,
                                          // buffer passa a ter "novo conteúdo"
  cout << buffer;                        // deve aparecer: novo conteúdo
  return 0;
}
```

2.

```
int main(){
  Mealheiro m1 {2, 1, 5, 10};
  m1.acrescenta(2);
  cout << m1.getAsString(); //apresenta: 1 moeda(s) de 1 centimo(s)
                            //           2 moeda(s) de 2 centimo(s)
                            //           1 moeda(s) de 5 centimo(s)
                            //           1 moeda(s) de 10 centimo(s)
                            //total      5 moeda(s) e 20 centimo(s)
  auto resultado = m1.retira(2); //retira todas as moedas de 2
  cout << m1.getAsString(); //apresenta: 1 moeda(s) de 1 centimo(s)
                            //           0 moeda(s) de 2 centimo(s)
                            //           1 moeda(s) de 5 centimo(s)
                            //           1 moeda(s) de 10 centimo(s)
                            //total      3 moeda(s) e 16 centimo(s)
  const Mealheiro m2 {10, 10};
  cout << m2.getAsString();//apresenta:  0 moeda(s) de 1 centimo(s)
                            //           0 moeda(s) de 2 centimo(s)
                            //           0 moeda(s) de 5 centimo(s)
                            //           2 moeda(s) de 10 centimo(s)
                            //total      2 moeda(s) e 20 centimo(s)
  return 0;
}
```

3.

```
class Data {
    int d, m, a;
public:
    Data(int d, int m, int a){ //Depois das validações necessárias
      this->d = d;
      this->m = m;
      this->a = a;
    }
     void setData(int d, int m, int a){ //Depois das validações necessárias
      this->d = d;
      this->m = m;
      this->a = a;
     }
};
class Funcionario{
     //dados do funcionario
public:
    Funcionario(){cout <<"Foi criado um funcionario"<<endl;}
    ~Funcionario(){cout <<"Foi destruido um funcionario"<<endl;}
};
class Animal{
     //dados do Animal - Implementar
public:
     //Construtor do Animal – Implementar
     //Criar um animal sendo uma cópia de outro – Implementar
     //getAsString função para ter o texto que representa o Animal – Implementar
     ~Animal(){ cout <<"Foi destruido um animal"<<endl;}
};
int main() {
    Funcionario avelino;
    Animal a("Tereco", "chimpanze", 10/2/2008, avelino); //id -> 1
    const Animal b("Bobby", "lobo", 31/12/2010, avelino); //id -> 2
    cout << a.getAsString();
    cout << b.getAsString();
    Animal c = b; //id do novo animal clonado -> 3
    cout << c.getAsString();
    //atenção: quando o programa terminar o Funcionario apenas é destruído uma
    //única vez
    return 0;
}
```