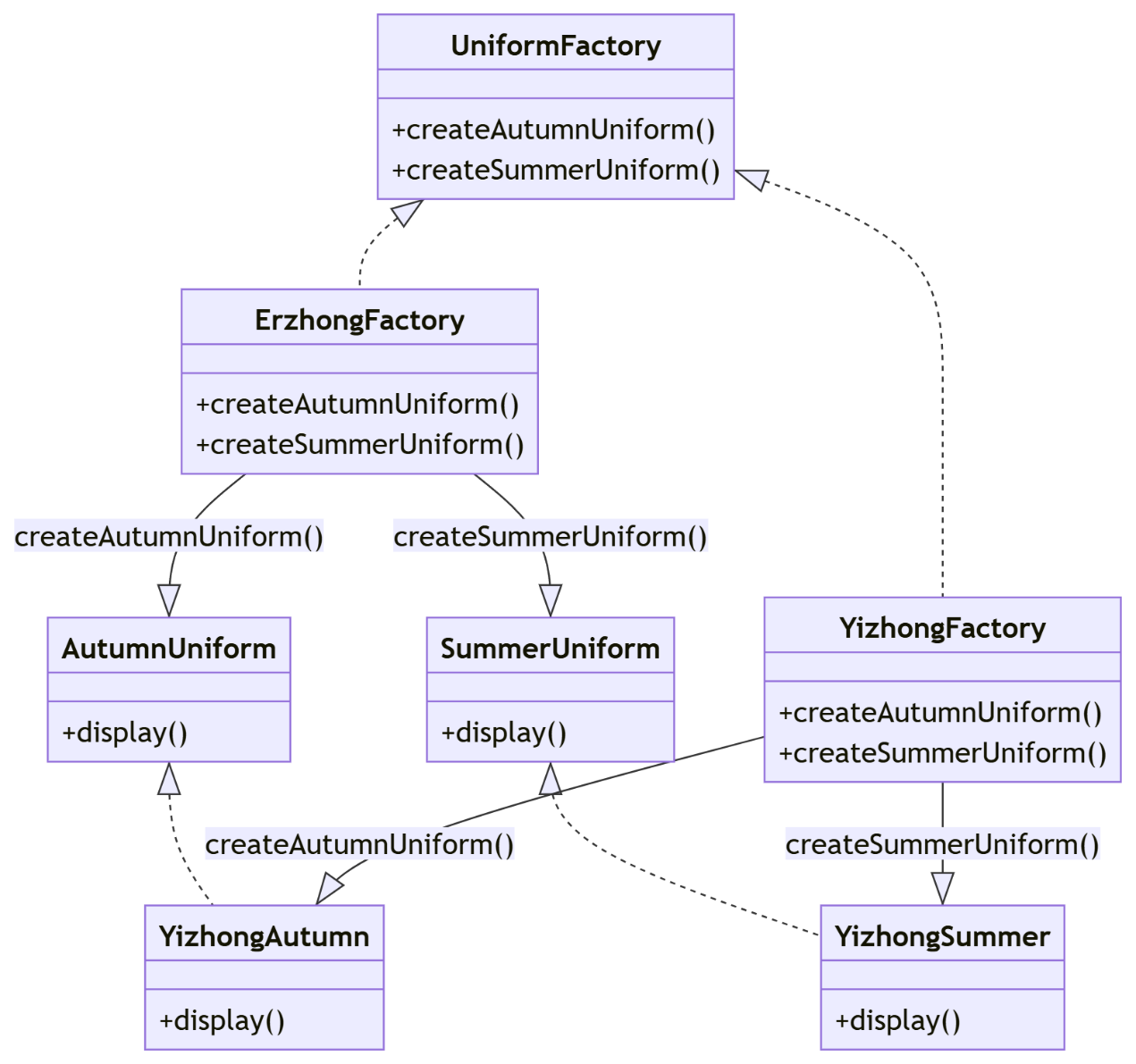
在这个场景中，最合适的设计模式是**抽象工厂模式**。使用抽象工厂模式可以创建一系列相关或依赖的对象，而无需指定它们的具体类。可以通过抽象工厂创建秋季和夏季的校服，并根据学校不同生成相应的校服套件。



|  |
| --- |
| **核心代码** |
| #include <iostream>  #include <string>  using namespace std;  // 抽象类：秋季校服  class AutumnUniform {  public:  virtual void display() = 0;  };  // 抽象类：夏季校服  class SummerUniform {  public:  virtual void display() = 0;  };  // 一中秋季校服  class YizhongAutumnUniform : public AutumnUniform {  public:  void display() override {  cout << "一中秋季校服：长袖上衣和秋季长裤" << endl;  }  };  // 一中夏季校服  class YizhongSummerUniform : public SummerUniform {  public:  void display() override {  cout << "一中夏季校服：短袖衬衣、短袖T恤、夏季长裤和短裤" << endl;  }  };  // 二中秋季校服  class ErzhongAutumnUniform : public AutumnUniform {  public:  void display() override {  cout << "二中秋季校服：长袖上衣和秋季长裤" << endl;  }  };  // 二中夏季校服  class ErzhongSummerUniform : public SummerUniform {  public:  void display() override {  cout << "二中夏季校服：短袖衬衣、短袖T恤、夏季长裤和短裤" << endl;  }  };  // 三中秋季校服  class SanzhongAutumnUniform : public AutumnUniform {  public:  void display() override {  cout << "三中秋季校服：长袖上衣和秋季长裤" << endl;  }  };  // 三中夏季校服  class SanzhongSummerUniform : public SummerUniform {  public:  void display() override {  cout << "三中夏季校服：短袖衬衣、短袖T恤、夏季长裤和短裤" << endl;  }  };  // 抽象工厂类：校服工厂  class UniformFactory {  public:  virtual AutumnUniform\* createAutumnUniform() = 0;  virtual SummerUniform\* createSummerUniform() = 0;  };  // 一中校服工厂  class YizhongFactory : public UniformFactory {  public:  AutumnUniform\* createAutumnUniform() override {  return new YizhongAutumnUniform();  }    SummerUniform\* createSummerUniform() override {  return new YizhongSummerUniform();  }  };  // 二中校服工厂  class ErzhongFactory : public UniformFactory {  public:  AutumnUniform\* createAutumnUniform() override {  return new ErzhongAutumnUniform();  }    SummerUniform\* createSummerUniform() override {  return new ErzhongSummerUniform();  }  };  // 三中校服工厂  class SanzhongFactory : public UniformFactory {  public:  AutumnUniform\* createAutumnUniform() override {  return new SanzhongAutumnUniform();  }    SummerUniform\* createSummerUniform() override {  return new SanzhongSummerUniform();  }  };  // 客户端代码：根据用户输入生产校服  int main() {  string school, season;  cout << "请输入学校名称（如：一中、二中、三中）和季节（如：秋季、夏季）：";  cin >> school >> season;  UniformFactory\* factory = nullptr;    // 根据输入选择工厂  if (school == "一中") {  factory = new YizhongFactory();  } else if (school == "二中") {  factory = new ErzhongFactory();  } else if (school == "三中") {  factory = new SanzhongFactory();  } else {  cout << "未知的学校！" << endl;  return -1;  }  // 根据输入选择生产秋季或夏季校服  if (season == "秋季") {  AutumnUniform\* autumnUniform = factory->createAutumnUniform();  autumnUniform->display();  delete autumnUniform;  } else if (season == "夏季") {  SummerUniform\* summerUniform = factory->createSummerUniform();  summerUniform->display();  delete summerUniform;  } else {  cout << "未知的季节！" << endl;  }  delete factory;  return 0;  } |