Adapter

The code below is meant to manage an investment portfolio using instances of class `Asset`. This implementation can be used to add assets such as real estate, but not `Bitcoin` (as `Bitcoin` is not a subclass of `Asset`). Using the appropriate design pattern, you are required to make the necessary changes so that Bitcoin could be included in an investment portfolio. Take note of the following:

- Classes `Asset`, `RealEstate`, `BitCoin`, and `Investor` should not be modified in any way.
- In addition to specifying the changes related to the design pattern, you need to complete the code that adds 1 Bitcoin ("BTC") to the portfolio in method `main`. Assuming the price of 1 BTC is $50000, the output should be 150000.

```
abstract class Asset{
        public abstract double getValue();
}

class RealEstate extends Asset{
        double value;

        public RealEstate(double value) {
                this.value = value;
        }

        @Override
        public double getValue() {
                return value;
        }
}

class Cryptocurrency{
        String code; //e.g. "BTC", "ETH"
        double amount;
        double rate; //price of 1 unit in CAD

        public Cryptocurrency(String code, double amount, double rate){
                this.code = code;
                this.amount = amount;
                this.rate = rate;
        }

        public double computeValue() {
                return amount * rate;
        }
}
```

```
class Investor{
	List<Asset> portfolio;

	public Investor() {
		portfolio = new ArrayList<Asset>();
	}

	public void addAsset(Asset asset) {
		portfolio.add(asset);
	}

	public double computePortfolioValue() {
		double result = 0;
		for(Asset asset:portfolio)
			result += asset.getValue();
		return result;
	}
}

public class Driver {
	public static void main(String [] args) {
		Investor investor = new Investor();
		investor.addAsset(new RealEstate(100000));
		//...
		//code that adds one Bitcoin to the portfolio
		//...
		System.out.println(investor.computePortfolioValue());
	}
}
```

Solution:

```
class Adapter extends Asset{
	Cryptocurrency crypto;

	public Adapter(Cryptocurrency crypto){
		this.crypto = crypto;
	}

	@Override
	public abstract double getValue(){
		return crypto.computeValue();
	}
}
```

In main:

```
//code that adds one Bitcoin to the portfolio
```

```
Cryptocurrency crypto = new Cryptocurrency(“BTC”, 1, 50000);
Adapter adapter = new Adapter(crypto);
investor.addAsset(adapter);
```